DRAFT

Cryptographic Communications System (CCS) Protection Profile

Version 1.06

21 November 2000

Prepared By: Booz·Allen & Hamilton

Prepared For: Department of Defense

DRAFT

Foreword

This Protection Profile (PP) was written by Booz·Allen and Hamilton, National Security Team, to support the Information Assurance Solutions Group. Please send comments on this PP to Brian Green, Erik Williams, or Jeff Kubik at Booz·Allen and Hamilton, 900 Elkridge Landing Road, Linthicum, MD 21090. This Protection Profile is the first formal submission by Booz·Allen and Hamilton to NSA on this particular concept of a high assurance cryptographic token. The PP was written to be compatible, as much as possible, with the Draft U.S. DoD Remote Access Protection Profile for High Assurance Environments, version 0.98, 24 May 2000 [1].

Table of Contents

List of Tables and Figures

Conventions and Terminology

Conventions

Except for replacing United Kingdom spelling with American spelling (at the client's request) the notation, formatting, and conventions used in this Protection Profile are consistent with version 2.1 of the Common Criteria (CC) [2] and the Draft U.S. DoD Remote Access Protection Profile for High Assurance Environments, version 0.98 (HARA PP). Selected presentation choices are discussed here to aid the Protection Profile (PP) user.

The CC allows several operations, defined in paragraph 2.1.4 of Part 2 of the CC, to be performed on functional requirements — refinement, selection, assignment, and iteration. Each of these operations is used in this Protection Profile. The refinement operation adds detail to a requirement, further restricting the requirement. **Bold text** denotes refinement of functional requirements. The selection operation is used to select one or more options provided by the CC in stating a requirement. The assignment operation is used to assign a specific value to an unspecified parameter, such as the length of a password. *Italicized text* denotes selections and assignments; however, one can determine which operation was performed by consulting the CC. From a specification viewpoint, only the words that result from the operation are important, not how the words were derived. Whenever a selection or assignment operation is left incomplete in this PP, it is offset with brackets ("[]") and the text *"ST selection"* or *"ST assignment,"* respectively, is indicated. These incomplete operations, along with their parameters, also appear in *italicized text*. In addition, there will be some unfilled "*ST assignment*" statements that may require reference to Appendix A for further information. The iteration operation specifies use of a component more than a single time. Multiple use of components may occur when an operation within the component must be completed multiple times (with differing values), or for different allocation of functions to sub-components within the TOE.

Application Notes are provided to help the developer, either to clarify the intent of a requirement, identify implementation choices, or to define "pass-fail" criteria for a requirement. For those components where Application Notes are appropriate, the Application Notes will follow the requirement component.

Terminology

In the Common Criteria, Section 2.3 of Part 1 defines many terms. In addition to terms defined in the Common Criteria, the authors have defined terms to aid the user of this Protection Profile. The CCS TOE consists of a High Assurance Cryptographic Token (HCT), a Communications Adapter (CA), and a Graphical User Interface (GUI) Application to access and manage the HCT.

audit administrator — An administrator who is authorized to view, backup, and delete the protected audit record or the role so defined. Note that audit administrators cannot have any other defined roles.

administrator — Any person that has the authority and responsibility for the long-term health of the security attributes of the system, or the defined role. Administrators can initiate, modify, view, and delete user security attributes. An administrator may have other roles, but no user may sign in with more than one role at a time.

agent — An individual that is not an authorized user of the TOE.

authorized users — Any person that is authorized to access the TOE and who has successfully authenticated to the TOE, or the defined role.

enclave — The secure, fixed facility that shelters and supports an IT environment on behalf of an organization. It contains an assortment of physical and electronic security mechanisms for authentication and access control.

external communication channels — Communication links between the HCT and the GUI components and communications between the CCS and other CCSs. Communications between the HCT and the GUI take place on the remote host hardware, facilitated by the remote host operating system, and because the remote host hardware and operating system components are not under the TOE scope of control, the communications between the HCT and GUI are defined as external communications. Similarly, communications between CCS units may occur on a variety of networks with different governing operating system and protocol combinations, and these communications are also considered external.

internal communication channel — Communication links within the HCT and between the HCT and the CA. Because these components within the CCS share an interface and because there are no other non-TOE products on which the communications depend, these communications are considered internal.

maintainer — A person who has a strictly limited time access to the TOE, generally for the purposes of resolving problems or performing preventive maintenance. This defined role or person requires more privileges than a user, but fewer privileges than an administrator.

object — An entity within the TSC that contains or receives information and upon which subjects perform operations.

remote host — Any computational device capable of generating, storing, processing, transmitting, and receiving information with the CCS using any approved IEEE interface. Physical access is controlled by the remote user.

remote users — An authorized user of the remote host.

subject — An entity within the TSC that causes operations to be performed.

system resources — Any system assets (data and software) required for the correct operation of the TOE.

TSF data — TOE Security Function data is information used by the TSF in making TOE Security Policy (TSP) decisions. TSF data may be influenced by users if allowed by the TSP. Security attributes, authentication data, and access control list entries are examples of TSF data.

unauthorized user — Any person that is not authorized, under the TSP, to access the TOE. This definition includes agents and authorized users who seek to exceed their authority.

user data — Data created by and for the authorized user, that does not affect the operation of the TSP. User data are the files that a user might upload or download to other remote units or the secure enclave. User data is separate from the TSF data, which has security attributes associated with it, and the system data.

user resources — Any data supplied by authorized users.

Document Organization

Section 1 provides the introductory material for the Protection Profile (PP).

Section 2 describes the Cryptographic Communications System (the TOE for this PP) and its general purpose.

Section 3 describes the expected environment for the CCS. This section also defines the set of threats that are to be addressed by either the technical countermeasures implemented in the CCS hardware or software or through the environmental controls.

Section 4 defines the security objectives for both the CCS and the CCS environment.

Section 5 contains the functional and assurance requirements derived from the Common Criteria, Parts 2 and 3, respectively, that must be satisfied by the CCS.

Section 6 provides a rationale to demonstrate explicitly that the information technology security objectives satisfy the policies and threats. Arguments are provided for the security objectives being necessary to support policies and counter threats. The section then explains how the set of requirements are sufficient to meet each objective, and that each security objective is addressed by one or more component requirements. Therefore, the two aforementioned subsections provide arguments that the security objectives and security requirements are both necessary and sufficient, respectively and collectively, to meet the needs dictated by the policies and threats. Next Section 6 provides a set of arguments that address dependency analysis, strength of function issues, and the internal consistency and mutual supportiveness of the protection profile requirements.

The reference section identifies background material.

An acronym list is provided to define frequently used acronyms.

1 – Introduction

Booz·Allen & Hamilton wrote this Protection Profile (PP) to support a DoD procurement of a Cryptographic Communications System (CCS). This PP is a component-level protection profile detailing the Policies, Threats, Assumptions, Security Objectives, Security Functional Requirements, and Security Assurance Requirements for the CCS component and its environment.

This PP will be of use to a few audiences: Information System Security Engineers (ISSEs), product vendors, and system integrators. Members of the primary audience include ISSEs designing secure information systems. The PP defines a minimal set of security requirements upon which specific implementations of the CCS can be specified, built, and tested. Also, vendors of CCS implementations will find this PP to be of value when they write their product Security Targets (STs). Finally, system integrators will find this PP useful for ensuring that the seamless integration of the CCS with the Remote Host and secure enclave components leads to an integrated solution that satisfies customer requirements.

The CCS PP team drew upon existing documentation that supports the High-Assurance Remote Access (HARA) Architecture, Version 1.1, 15 May 2000 [3], the HARA Protection Profile [2], and the existing solution developed by the Remote Access Security Program (RASP). However, every attempt was made to make the PP implementation independent to maximize flexibility for innovation and minimize design constraints.

The frame of this Protection Profile was written using the CCToolbox, but the CCS team felt that a significant amount of work remained to be done before the CCToolbox output would be fit to serve the customer's needs.

1.1 - Identification

Title:	Cryptographic Communication System
Authors:	Erik Williams, BA&H; Jeff Kubik, BA&H; Angus Forbes, Litton TASC
Vetting Status:	Final Draft
CC Version:	2.1 Final
Evaluation Level:	EAL 5, augmented
General Status:	Active
Registration:	TBD
Keywords:	high assurance, communications, remote access, cryptographic token

1.2 - Protection Profile Overview

This Protection Profile specifies the DoD's information security needs for the CCS component. The communications media (telephone network, wireless network, Internet connection) for remote access may be outside the sphere of ownership and management of the enterprise making the remote connection. More details are provided in Section 2 of this Protection Profile. This PP specifies the Policies, Threats, Assumptions, Security Objectives, Security Functional Requirements, and Security Assurance Requirements for the CCS component and its environment. There is a strong dependence on functionality from the environment because components depend on their system surroundings to produce many useful and secure functions.

1.3 - Related Protection Profiles

This Protection Profile is based on the HARA PP (Draft U.S. DoD Remote Access Protection Profile for High Assurance Environments, Version 0.98, May 2000).

2 - TOE Description

The Cryptographic Communications System (CCS) provides in-line data encryption capabilities to enable traveling or telecommuting users to securely access their local LANs, enclaves, or enterprise-computing environments via commercial common carrier networks. These networks may include the public switched telephone network (PSTN), the Internet or other packet switched networks, and wireless networks. The initial connection to the commercial common carrier may take place in a foreign country, and the network may be foreign government owned.

The CCS consists of a High Assurance Cryptographic Token (HCT), a Communications Adapter (CA), and a Graphical User Interface (GUI) Application to access and manage the HCT. The communication network through which the communications channel is created is not trusted and may be shared with hostile users. The remote user's computing assets (hardware and operating system) are not evaluated and are physically vulnerable. The remote user must share responsibility for protecting the assets with the protective technology on the assets. The CCS will only operate when plugged into a remote host. In particular, the HCT in the CCS should be unclassified when no users are logged onto it. Security policy will determine if the CCS can be left unattended.

This Protection Profile supports the scenario traveling or telecommuting users in high-risk environments accessing remotely information that requires high-assurance protection. The set of traveling users includes government and business personnel anywhere in the world and telecommuters connecting to select secure enclaves and other remote hosts worldwide. Under all circumstances, the user should know when security features are enabled, and more importantly, when they are not.

Figure 1 depicts the components comprising the CCS – the High Assurance Cryptographic Token (HCT), the Communications Adapter (CA) and the Graphical User Interface (GUI) application. The figure exhibits the theory of operation for the remote unit using the following scenario:

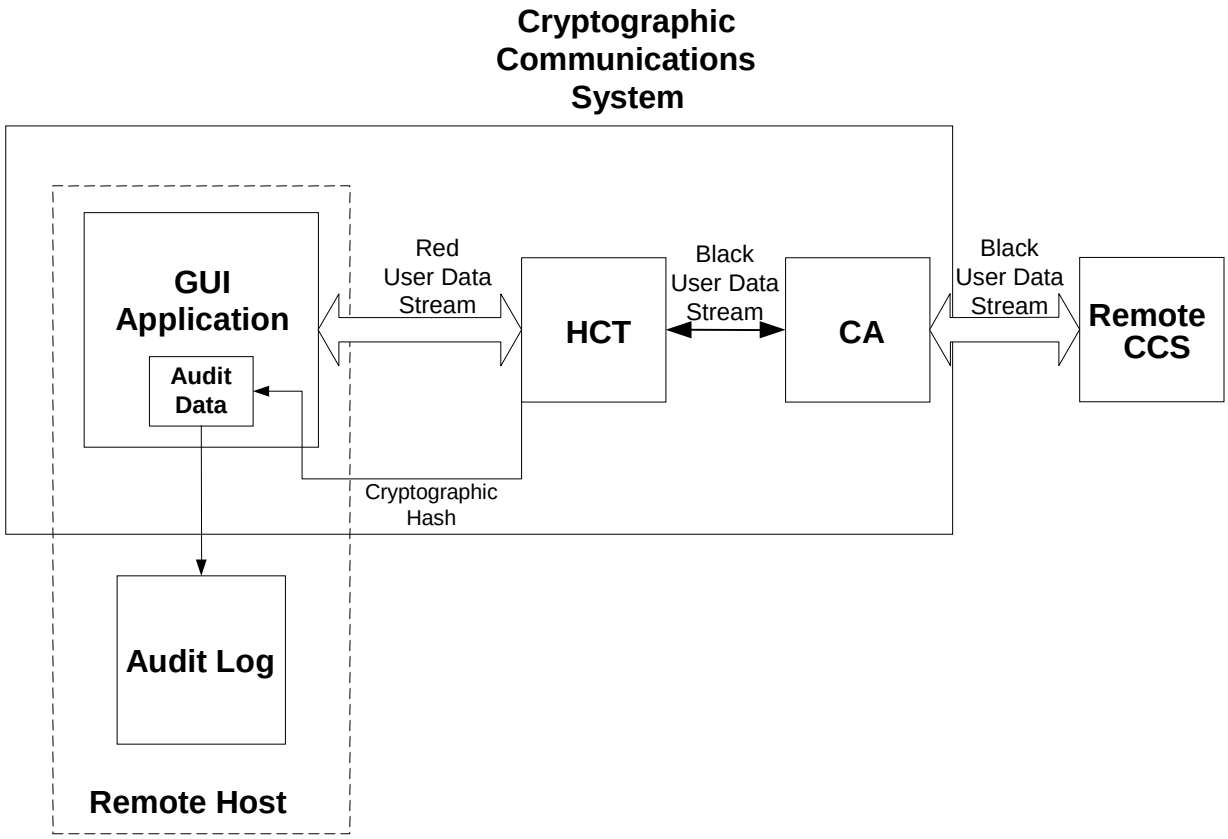

Figure 1: Cryptographic Communications System

When the user wants to communicate with the secure enclave or another remote host, he or she has to log into the HCT via the GUI on the remote host platform. The GUI software that runs on the user's remote host provides the interface for user identification and authentication (I&A) and supports audit data generation. The HCT component of the CCS provides a cryptographic hash of the audit data to deter and detect audit log access and tampering. The GUI software includes modules limited to the use of system security administrators for account and audit log maintenance. The CCS provides in-line encryption, key management, and authentication algorithms and protects itself from tampering and other technical attacks. The CCS shall incorporate traffic flow security functions such that, once the communication link is established, message metadata is obscured. The CCS provides all critical security services offered locally to the remote host for protecting the external communications links and message contents. Communications between the remote host and the enclave or other remote host are enabled if and only if there is mutual recognition between the HCTs at both end-points. Various CAs provide different communications interfaces to the HCT. The key-fill CA is dedicated to key-fill purposes only. Remote key updates are allowed through the key-fill CA. Additional IEEE approved data I/O ports may be integrated into the CCS later. Information drawn from either the secure enclave or other remote host must be decrypted by the HCT.

Not shown in the figure, are the roles and responsibilities of the audit administrator, administrator, and maintainer. The audit administrator is authorized to view, backup, and delete the protected audit record. The administrator has the authority and responsibility for the long-term health of the security attributes of the system. Audit administrators may not have any dual roles as administrators, maintainers, or users. Administrators can initiate, modify, view, and delete user, maintainer, administrator, and audit administrator security attributes. The maintenance person has short-term (task time limited to resolving problems and providing preventive maintenance) access privileges (limited to those required to resolve problems and provide preventive maintenance) that are higher than users but lower than those of administrators. Users are authorized to use the CCS and remote host to accomplish their mission.

3 - TOE Security Environment

The security environment of the traveling users and telecommuters connecting to selected secure enclaves worldwide is highly variable. Traveling users may be in very high-risk, physically hostile environments when communicating with a secure enclave or other remote hosts. Telecommuters may communicate from the same location several times per day in the relative safety of his or her U.S. home. The communications network or connection may be owned or operated by an adversary or by an American communications company. The remote host and operating system may be commercially available, non-evaluated products made in an adversarial country. The number of sessions may be as few as one, but there may be no upper limit to the number of times a user may try to connect and communicate with the enclave. The length of transmission could be all day for the telecommuter, or it may be very short in a hostile tactical environment.

Although the CCS is designed as an independent component, it cannot operate without direct connections with a remote host. This protection profile is concerned neither with any specific remote host hardware platform nor with a particular operating system. A separate protection profile will address the remote host as a TOE. Therefore, any intrusion detection, anti-virus, or audit storage capabilities will reside within the remote host's security requirements, not the CCS. Many of these non-TOE requirements for the CCS would serve as potential TOE requirements for the remote host protection profile.

This Protection Profile supports the scenario of remote access in a high-risk environment to information requiring high-assurance protection. Within this scenario, an authorized user is cleared to access all information within the enclave but may not have the required "need-to-know." Consequently, this PP does not address multi-level security requirements.

Chapter 3 describes the Assumptions, Threats, and Policies that are relevant to both the CCS TOE and the CCS TOE environment. The first section describes the Secure Usage Assumptions — these are the assumptions that support secure use of the CCS. Assumptions generally support achieving Security Objectives by eliminating some concerns. Threats are countered by the Security Objectives. Policies support the Security Objectives and are employed by Security Objectives to counter Threats. Author generated Policies, Threats, and Assumptions are listed in ALL-CAPS.

3.1 - Secure Usage Assumptions

Because the TOE is the CCS component, there are more Secure Usage Assumptions that this PP depends on relative to the HARA System PP. Assumptions are limiting conditions that are accepted before developing policy or considering threats.

A.CRYPTO: The strength of functions and management procedures required by the US Department of Defense in OE.CRYPTO_DESIGN and O.CRYPTO_OPNS are suitable for their intended use.

A.OPERATING_SYS: The Operating System functions as an intermediary between components within the Remote Host. The Operating System does not have vulnerabilities that undermine the secure operation of the TOE. The security requirements for the Operating System are not defined in this PP.

A.PEER: Any other systems with which the TOE communicates are assumed to be under the same management control and operate under the same security policy constraints. Other systems include other remote hosts, secure enclaves, and dedicated key-fill devices, which are trusted to supply keys that meet the requirements of A.CRYPTO. At no time will a remote host access data or use key at a different classification level than its current operational level.

A.TEMPEST: Emanations from the TOE will meet customer requirements in P.TEMPEST.

Application Note: Formal TEMPEST requirements do not typically appear in a protection profiles because the Common Criteria does not address TEMPEST; therefore NIAP labs cannot perform TEMPEST testing as part of their Common Criteria evaluation. Thus, there needs to be A.TEMPEST to satisfy P.TEMPEST within the TOE.

A.TRUSTED_ADMIN: Administrators are trusted, competent, and trained. Administrators are trusted to follow policies and procedures defined in the TOE for secure administration of the TOE and perform their duties in a manner and that does not compromise the security of the TOE most of the time — i.e. mistakes may happen, but not often, and not with bad intentions.

A.TRUSTED_USER: Users are trusted, competent, and trained. Users are trusted to follow policies and procedures defined in the TOE for secure administration of the TOE and perform their duties in a manner and that does not compromise the security of the TOE most of the time — i.e. mistakes may happen, but not often, and not with bad intentions.

3.2 - Threats to Security

The threats listed here are general threats. The detailed attacks listed in the CCToolbox are not listed in this PP. The authors considered that countering the particular detailed attacks were left more appropriately to Security Target authors. Threats are actions that may have an adverse affect on the CCS, remote host, or mission.

T.ALTER: An unauthorized user may surreptitiously gain access to the TOE and attempt to alter, replace, and/or deny access to system elements (e.g. hardware, firmware, or software) in an attempt to subvert the device.

T.Component_Failure: Failure of one or more system components results in the loss of system-critical functionality.

T.CRASH: Due to interruption of the operation of the TOE resulting from power failure or other unforeseen interruptions, security critical information is either incomplete or corrupted.

T.Dev_Flawed_Code: A system or applications developer delivers code that does not perform according to specifications or contains security flaws.

T.ERROR: An authorized user or administrator may perform erroneous actions that will compromise user and/or system resources.

T.Hack_AC: A hacker gains undetected access to a system due to missing, weak and/or incorrectly implemented access control causing potential violations of integrity, confidentiality, or availability.

T.Hack_Crypto: A hacker performs cryptanalysis on encrypted data in order to recover message content.

T.Hack_Masq: A hacker masquerades as an authorized user to perform operations that will be attributed to the authorized user or a system process.

T.HACK_TRAFFIC: A hacker or an eavesdropper performs traffic analysis on message traffic to gather intelligence (e.g. indicators or warnings of the intentions of the person or organization sending or receiving messages).

T.IMPORT: An authorized user, administrator, and/or remote IT system of the TOE may unwittingly introduce malicious code into the system, resulting in a compromise of the integrity, availability, and/or confidentiality of user and/or system resources.

T.PHYSICAL: Security-critical parts of the TOE may be subject to physical attack by agents to compromise security.

3.3 - Organizational Security Policies

The policies listed here are general policies. The detailed policies listed in the CCToolbox are not listed in this PP. The authors considered that supporting the particular detailed policies were left more appropriately to Security Target authors for the particular organization affected. After policymakers consider relevant threats and acknowledge their Assumptions, they create Policies that describe the organizational actions, consequences, people, and circumstances surrounding the use of the CCS.

P.ACCOUNT: User activity shall be monitored so that they may be held accountable for their actions, sanctions can be applied when malfeasance occurs, and proper application of system controls is ensured. All users will be notified that such monitoring may occur.

P.Authorities: Appropriate authorities shall be immediately notified of any threats or vulnerabilities affecting systems that process their data.

P.Authorized_Use: Information shall be used only for its authorized purpose(s).

P.Availability: Information shall be available to satisfy mission requirements.

P.CONFIDENTIALITY: The confidentiality of user and system data stored or processed in the TOE must be protected.

P.Guidance: Guidance shall be provided for the secure installation, administration, and use of the system.

P.Information_AC: Only authorized individuals and processes shall access information.

P.INTEG: The integrity of user and system data stored or processed in the TOE must be protected.

P.Lifecycle: Information systems security shall be an integral part of all system lifecycle phases.

P.MANAGE: The TOE shall be managed such that its security functions are implemented and preserved throughout its operational lifetime.

P.Physical_Control: Information shall be physically protected to prevent unauthorized disclosure, destruction, or modification.

P.TEMPEST: The TOE shall be constructed such that all emanations of red or data satisfy the customer TEMPEST design requirements.

4 - Security Objectives

4.1 - Security Objectives for the TOE

The objectives written in ALL-CAPS correspond with the objectives in the higher level HARA document; however, the CCS PP required that the PP authors specify objectives that did not neatly fit into the objectives of the higher level HARA PP. The CCS-specific objectives (relative to the HARA PP) are written in Title Case.

O.ACCESS: The TOE will control access to information that is subject to the TOE security policy, based on the identity of the individuals, such that this policy cannot be bypassed in the TOE. The TOE will restrict the actions a user may perform before the TOE verifies the identity of the user and will provide mechanisms to limit the number of user initiated sessions open at one time.

O.AUDIT: The TOE will provide support for an audit trail to ensure each authenticated user and TOE administrator can be held accountable for his or her actions in the TOE. The audit trail will be of sufficient detail to reconstruct events in determining the cause or magnitude of compromise should a security violation or malfunction occur. The TOE will support the collection of an audit trail. It will not maintain that trail nor will it perform queries on the trail to generate reports. It may be possible to configure the TOE to report varying levels of audit data based on the audit policy in effect for a particular user. Those varying levels may also be adjusted based on the time of day, day of the week, duration of a trip or mission, etc. The audit function will display to the authenticated user the most recent successful and unsuccessful attempts to establish a session as the user. The TOE will deter modification or destruction of audit data through the creation of an audit administrator role. The TOE will use a cryptographic hash on the audit data to detect and deter tampering. The audit log will uniquely identify each user and record the date and time of action, action, the subject performing the action, and the object acted upon.

O.CRYPTO_OPNS: The TOE will support cryptographic functions in a secure manner. User access to cryptographic IT assets will be restricted in accordance with a specified user access control policy. The TOE will provide one or more roles to manage cryptographic assets and attributes. There will be complete separation provided between plaintext and encrypted data and between data and keys. This requires separate channels and separate storage areas for data and keys. The only place any data can pass between the plaintext and encrypted data modules is in the cryptographic engine. There should be no way for plaintext keys to reach the data module and no way for data to enter the key-handling module. Encrypted keys can be handled as encrypted data, but with limited user access. The TOE will protect cryptographic data assets when they are being transmitted to and from the TOE, either through intervening untrusted components or directly to/from human users. The TOE will prohibit the transmission of a ciphertext message over internal circuitry where the corresponding plaintext might be available. To protect message metadata, the TOE will support cryptographic padding (e.g. random plaintext) and encrypted addressing. The cryptographic components, functions, and interfaces shall be fully defined to ensure that the cryptographic keys have appropriate protection

throughout their lifecycle, including generation, distribution, storage, use, and destruction. There will be self-tests, as well as alarms, alarm checks, and redundant logic, to provide the ability to verify that the cryptographic functions operate as designed. The TOE will produce, through robust encryption techniques, cipher text that cannot be decrypted without either massive computational power or knowledge of the encryption key.

O.Data_Exchange_Conf: The TOE will protect user data confidentiality when exchanging data with a remote system.

O.FAULT_TOLERANT: The TOE will provide fault tolerant operations for critical components and continue to operate in the presence of specific failures in one or more system components. The TOE will automatically recover to a secure state without security compromise after system error or other interruption of system operation. The TOE will preserve the secure state of the system, as well as the level of assurance of the system, in the event of a secure component failure.

O.IDENTIFY: The TOE will uniquely identify and authenticate each user of the system to support accountability through basic I&A functions. The TOE will associate each user-requested action with the identity of the user who initiated the session (i.e., log on).

O.INTEGRITY: The TOE will provide the following technical features to protect its system security functions: detect changes to its security-related functions and user data, protect against tampering by users, and protect against attempts by users to bypass its security functions. The TOE will provide the ability for authorized users to verify that the system operates as designed, to conduct periodic integrity checks on both system and user data, and to conduct periodic system functional tests to test the integrity of the hardware and code running system functions. The TOE will always invoke mechanisms that enforce security policies. It will maintain at least one security domain for system (TOE) execution to protect the TOE from interference and tampering. Likewise, it will ensure that security-relevant software, hardware, and firmware are correctly functioning through features and procedures. The TOE will provide system features that detect physical tampering of a system component and will use those features to limit security breaches. The TOE will send integrity data, results of the integrity checks, to the audit trail. Additionally, it will prevent or resist physical tampering with specified system devices and components.

O.Integrity_Attr_Exch: The TOE will ensure that the system correctly exchanges security-attribute information with another trusted IT product.

O.MANAGE: The TOE will provide adequate management features for its security functions. It will maintain security-relevant roles and the association of users with those roles. In addition to user identity, the TOE will maintain a set of security attributes associated with individual users. The TOE will provide features to specify object classes (domains), user groups, and operation classes. The management features will control what users can do in a given group by specifying which users may perform certain operations on particular objects.

O.No_Residual_Info: The TOE will ensure there is no "object reuse," i.e., ensure that there is no residual information in some information containers or system resources upon their reallocation to different users.

O.Session_Termination: The TOE will lock and then terminate a session after a given interval of inactivity.

4.2 - Security Objectives for the Environment

OE.ADMIN: Administrators manage the TOE in a manner that maintains the system security. While the TOE is in operation, the administrator will control access to the system by maintenance personnel who troubleshoot the system and perform system updates. To securely manage the TOE, the administrator should know the origin of all data files and executables that the TOE, remote host, and secure enclave may generate, store, process, transmit, or receive. Administrators will terminate maintenance user system access privileges after expiration of an assigned timed interval. The administrator will also manage the initialization of, values for, and limits on allowable operations on security attributes, security critical data, and security mechanisms. The administrator, using the security tools and techniques employed during the development phase, will detect and resolve flaws during the operational phase and document the flaws. When TOE hardware, software, or firmware must be destroyed, the administrator will employ safe destruction techniques. Administrators will apply code fixes to fix the code when there are known security vulnerabilities in the code. This is particularly important with respect to the operating system. The administrator will implement a configuration management plan to assure storage integrity, identify system connections, and identify the system components (software, hardware, and firmware). Part of configuration management is ensuring that integrity data is not lost or misplaced. Any circumstances that can cause untrusted recovery will be documented with mitigating procedures established. Configuration management is critical to maintaining certification to operate the TOE. The administrator will manage and update user authorization and privilege data, system security policy data, enforcement functions, and other security-relevant configuration data in accordance with organizational security policies. Administrators are responsible for the proper disposal of user data after access removal (due to job termination etc.). The administrator will manage resource security attributes and security-critical (TSF) data to ensure that the size of the data does not exceed the space allocated for storage of the data. The administrator will communicate system threats and vulnerabilities to system stakeholders.

OE.AUDIT_MAINTAIN: Administrators will apply technical, procedural, and administrative controls that are sufficient to maintain user accountability throughout the TOE. An audit-administration role will be created. The audit administrator will define the system response to possible loss of audit records when audit trail storage is full or nearly full; protect audit records against unauthorized access, modification, or deletion to ensure accountability of user actions; maintain audit data, guarantee space for that data, and regularly review audit data. The administrator will communicate anomalous audit data to system stakeholders.

OE.BANNER: The system will provide a banner to notify all users that they are entering a government or business computer system and their actions will be audited. Consequently, the banner informs the user of the possibility that the system will monitor user actions and that misuse of the system may result in criminal, civil, or administrative penalties.

OE.CRYPTO_DESIGN: This objective implements the engineering design requirements that DoD imposes on cryptosystems. The developer shall fully define cryptographic components, functions, and interfaces; minimize, or even eliminate, design and implementation errors in the cryptographic modules and functions, and prevent errors in one part of the TOE from influencing other parts, especially cryptographic parts. To this end, non-cryptographic input/output paths must be well defined and logically independent of circuitry and processes performing key generation, manual key entry, key erasure, and similar key-related operations. The developer shall specify cryptographic security functional requirements (SFRs) that are expected to be handled by other software, hardware, or firmware that is external to the TOE. The developer shall test cryptographic operation and key management functions.

OE.Malicious_Code: Administrators will incorporate malicious code prevention procedures and mechanisms.

OE.OPERATE: Authorized users and administrators will operate the TOE in a manner that maintains the system security by following adequate guidance documentation. Documentation provided to them will detail the proper use of the TOE to minimize the security risks within the environment.

OE.Screen_Lock: The operating system or an application will provide a screen lock function to prevent an unauthorized user from using an unattended computer where a valid user has an active session.

OE.Source_Code_Exam: The developer, an independent tester, or an administrator (or a combination of all three parties) will examine source code for accidental or deliberate flaws in code made by the developer. The accidental flaws could be lack of engineering detail or bad design. Deliberate flaws would include building trapdoors for later entry.

5 - IT Security Requirements

Chapter 5 presents the iteration, assignment, selection, and refinement statements of the security functional and assurance requirements. Section 5.1 defines the security functional requirements for the TOE. Section 5.2 directs the reader to other sections because no security assurance requirements were selected to be implemented in the TOE. Section 5.3 details the security functional and assurance requirements for the environment.

5.1 - TOE Security Functional Requirements

5.1.1 - Audit data generation (FAU_GEN.1)

FAU_GEN.1.1 - The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions;
b) All auditable events for the [*detailed*] level of audit; and
c) [*ST assignment: other specifically defined auditable events*].

FAU_GEN.1.2 - The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*ST assignment: other audit relevant information*]

5.1.2 - User identity association (FAU_GEN.2)

FAU_GEN.2.1 - The TSF shall be able to associate each auditable event with the identity of the user that caused the event.

5.1.3 - Selective audit (FAU_SEL.1)

FAU_SEL.1.1 - The TSF shall be able to include or exclude auditable events from the set of audited events based on the following attributes:

a) [*ST selection: object identity, user identity, subject identity, host identity, event type*];
b) [*ST assignment: list of additional attributes that audit selectivity is based upon*].

5.1.4 - Cryptographic key generation (FCS_CKM.1)

FCS_CKM.1.1 - The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [*ST assignment: cryptographic key generation algorithm*] and specified cryptographic key sizes [*ST assignment: cryptographic key sizes*] that meet the following: [*ST assignment: list of standards*].

Note: The ST customer should provide the development vendor specific information to complete these assignments.

5.1.5 - Cryptographic key distribution (FCS_CKM.2)

FCS_CKM.2.1 - The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method [*assignment: cryptographic key distribution method*] that meets the following: [*assignment: list of standards*].

5.1.6 - Cryptographic key access (FCS_CKM.3)

FCS_CKM.3.1 - The TSF shall perform [*assignment: type of cryptographic key access*] in accordance with a specified cryptographic key access method [*assignment: cryptographic key access method*] that meets the following: [*assignment: list of standards*].

Note: The ST customer should provide the development vendor specific information to complete these assignments.

5.1.7 - Cryptographic key destruction (FCS_CKM.4)

FCS_CKM.4.1 - The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [*assignment: cryptographic key destruction method*] that meets the following: [*assignment: list of standards*].

Note: The ST customer should provide the development vendor specific information to complete these assignments.

5.1.8 - Cryptographic operation (FCS_COP.1)

FCS_COP.1.1 - The TSF shall perform [*assignment: list of cryptographic operations*] in accordance with a specified cryptographic algorithm [*assignment: cryptographic algorithm*] and cryptographic key sizes [*assignment: cryptographic key sizes*] that meet the following: [*assignment: list of standards*].

Note: The ST customer should provide the development vendor specific information to complete these assignments.

5.1.9 - Subset access control (FDP_ACC.1)

FDP_ACC.1.1 - The TSF shall enforce the [*ST assignment: access control SFP*] on [*ST assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP*].

5.1.10 - Security attribute based access control (FDP_ACF.1)

FDP_ACF.1.1 - The TSF shall enforce the [*ST assignment: access control SFP*] to objects based on [*ST assignment: security attributes, named groups of security attributes*].

FDP_ACF.1.2 - The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [*ST assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*].

FDP_ACF.1.3 - The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [*ST assignment: rules, based on security attributes, that explicitly authorize access of subjects to objects*].

FDP_ACF.1.4 - The TSF shall explicitly deny access of subjects to objects based on the [*ST assignment: rules, based on security attributes, that explicitly deny access of subjects to objects*].

5.1.11 - Export of user data without security attributes (FDP_ETC.1)

FDP_ETC.1.1 - The TSF shall enforce the [*ST assignment: access control SFP(s) and/or information flow control SFP(s)*] when exporting user data, controlled under the SFP(s), outside of the TSC.

FDP_ETC.1.2 - The TSF shall export the user data without the user data's associated security attributes.

5.1.12 - Subset information flow control (FDP_IFC.1)

FDP_IFC.1.1 - The TSF shall enforce the [*ST assignment: information flow control SFP*] on [*ST assignment: list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP*].

5.1.13 - Simple security attributes (FDP_IFF.1)

FDP_IFF.1.1 - The TSF shall enforce the [*ST assignment: information flow control SFP*] based on the following types of subject and information security attributes: [*ST assignment: the minimum number and type of security attributes*].

FDP_IFF.1.2 - The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: [*ST assignment: for each operation, the security attribute-based relationship that must hold between subject and information security attributes*].

FDP_IFF.1.3 - The TSF shall enforce the [*ST assignment: additional information flow control SFP rules*].

FDP_IFF.1.4 - The TSF shall provide the following [*ST assignment: list of additional SFP capabilities*].

FDP_IFF.1.5 - The TSF shall explicitly authorize an information flow based on the following rules: [*ST assignment: rules, based on security attributes, that explicitly authorize information flows*].

FDP_IFF.1.6 - The TSF shall explicitly deny an information flow based on the following rules: [*ST assignment: rules, based on security attributes, that explicitly deny information flows*].

5.1.14 - No illicit information flows (FDP_IFF.5)

FDP_IFF.5.1 - The TSF shall ensure that no illicit information flows exist to circumvent [*ST assignment: name of information flow control SFP*].

5.1.15 - Import of user data without security attributes (FDP_ITC.1)

FDP_ITC.1.1 The TSF shall enforce the [*ST assignment: access control SFP and/or information flow control SFP*] when importing user data, controlled under the SFP, from outside of the TSC.

FDP_ITC.1.2 The TSF shall ignore any security attributes associated with the user data when imported from outside the TSC.

FDP_ITC.1.3 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TSC: [*ST assignment: additional importation control rules*].

5.1.16 - Full residual information protection (FDP_RIP.2)

FDP_RIP.2.1 - The TSF shall ensure that any previous information content of a resource is made unavailable upon the [*ST selection one or both: allocation of the resource to, de-allocation of the resource from*] all objects.

5.1.17 - Stored data integrity monitoring (FDP_SDI.1)

FDP_SDI.1.1 - The TSF shall monitor user data stored within the TSC for [*integrity errors*] on all objects, based on the following attributes: [*user data attributes*].

5.1.18 - Basic data exchange confidentiality (FDP_UCT.1)

FDP_UCT.1.1 - The TSF shall enforce the [*access control SFP(s) and/or information flow control SFP(s)*] to be able to [*transmit, receive*] objects in a manner protected from unauthorized disclosure.

5.1.19 - Data exchange integrity (FDP_UIT.1)

FDP_UIT.1.1 The TSF shall enforce the [access control SFP(s) and/or information flow control SFP(s)] to be able to [transmit, receive] user data in a manner protected from [modification, deletion, insertion, replay] errors.

FDP_UIT.1.2 The TSF shall be able to determine on receipt of user data, whether [*modification, deletion, insertion, replay*] has occurred.

5.1.20 - Authentication failure handling (FIA_AFL.1)

FIA_AFL.1.1 - The TSF shall detect when [*ST or NSA assignment: number*] unsuccessful authentication attempts occur related to [*HCT PIN entry*].

FIA_AFL.1.2 - When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [*lock out the HCT until full RU reboot*].

5.1.21 - User attribute definition (FIA_ATD.1)

FIA_ATD.1.1 - The TSF shall maintain the following list of security attributes belonging to individual users: [*ST assignment: list of security attributes*].

5.1.22 - User authentication before any action (FIA_UAU.2)

FIA_UAU.2.1 - The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

5.1.23 - Protected authentication feedback (FIA_UAU.7)

FIA_UAU.7.1 - The TSF shall provide only [*asterisks or other symbolic characters that do not resemble or reveal the password*] to the user while the authentication is in progress.

Note: The user will receive a message that the authentication process is proceeding, “please wait…”

5.1.24 - User identification before any action (FIA_UID.2)

FIA_UID.2.1 - The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

5.1.25 - User-subject binding (FIA_USB.1)

FIA_USB.1.1 - The TSF shall associate the appropriate user security attributes with subjects acting on behalf of that user.

5.1.26 - Secure security attributes (FMT_MSA.2)

FMT_MSA.2.1 - The TSF shall ensure that only secure values are accepted for security attributes.

5.1.27 - Static attribute initialization (FMT_MSA.3)

FMT_MSA.3.1 - The TSF shall enforce the [*ST assignment: access control SFP, information flow control SFP*] to provide [*restrictive*] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 - The TSF shall allow the [*administrator*] to specify alternative initial values to override the default values when an object or information is created.

5.1.28 - Management of TSF data (FMT_MTD.1)

FMT_MTD.1.1 - The TSF shall restrict the ability to [*change default, query, modify, delete*][*ST assignment: other operations*] the [*ST assignment: list of TSF data*] to [*administrator*].

5.1.29 - Restrictions on security roles (FMT_SMR.2)

FMT_SMR.2.1 - The TSF shall maintain the roles: [*the authorized identified roles of user and administrator*].

FMT_SMR.2.2 - The TSF shall be able to associate users with roles.

FMT_SMR.2.3 - The TSF shall ensure that the conditions [ST *assignment: conditions for the different roles*] are satisfied.

5.1.30 - Assuming roles (FMT_SMR.3)

FMT_SMR.3.1 - The TSF shall require an explicit request to assume the following roles: [*assignment: the roles: audit administrator, administrator, maintainer*].

5.1.31 - Abstract machine testing (FPT_AMT.1)

FPT_AMT.1.1 - The TSF shall run a suite of tests [*ST selection: during initial start-up, periodically during normal operation, at the request of an authorized user, other*

conditions] to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.

5.1.32 - Failure with preservation of secure state (FPT_FLS.1)

FPT_FLS.1.1 - The TSF shall preserve a secure state when the following types of failures occur: [*ST assignment: list of types of failures in the TSF*].

5.1.33 - Passive detection of physical attack (FPT_PHP.1)

FPT_PHP.1.1 - The TSF shall provide unambiguous detection of physical tampering that might compromise the TSF.

FPT_PHP.1.2 - The TSF shall provide the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred.

5.1.34 - Resistance to physical attack (FPT_PHP.3)

FPT_PHP.3.1 - The TSF shall resist *[ST assignment: physical tampering scenarios*] to the [*ST assignment: list of TSF devices/elements*] by responding automatically such that the TSP is not violated.

5.1.35 - Manual recovery (FPT_RCV.1)

FPT_RCV.1.1 - After a failure or service discontinuity, the TSF shall enter a maintenance mode where the ability to return the TOE to a secure state is provided.

5.1.36 - Function recovery (FPT_RCV.4)

FPT_RCV.4.1 - The TSF shall ensure that [*ST assignment: list of SFs and failure scenarios*] have the property that the SF either completes successfully, or for the indicated failure scenarios, recovers to a consistent and secure state.

5.1.37- Non-bypassability of the TSP (FPT_RVM.1)

FPT_RVM.1.1 - The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

5.1.38 - Complete reference monitor (FPT_SEP.3)

FPT_SEP.3.1 - The unisolated portion of the TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.3.2 - The TSF shall enforce separation between the security domains of subjects in the TSC.

FPT_SEP.3.3 - The TSF shall maintain the part of the TSF that enforces the access control and/or information flow control SFPs in a security domain for its own execution that protects them from interference and tampering by the remainder of the TSF and by subjects untrusted with respect to the TSP.

5.1.39 - Reliable time stamps (FPT_STM.1)

FPT_STM.1.1 - The TSF shall be able to provide reliable time stamps for its own use.

5.1.40 - Inter-TSF basic TSF data consistency (FPT_TDC.1)

FPT_TDC.1.1 - The TSF shall provide the capability to consistently interpret [*ST assignment: list of TSF data types*] when shared between the TSF and another trusted IT product.

FPT_TDC.1.2 - The TSF shall use [*ST assignment: list of interpretation rules to be applied by the TSF*] when interpreting the TSF data from another trusted IT product.

5.1.41 - TSF testing (FPT_TST.1)

FPT_TST.1.1 - The TSF shall run a suite of self tests [*ST selection: during initial start-up, periodically during normal operation, at the request of the authorized user, at the conditions*][*assignment: conditions under which self test should occur*] to demonstrate the correct operation of the TSF.

FPT_TST.1.2 - The TSF shall provide authorized users with the capability to verify the integrity of TSF data.

FPT_TST.1.3 - The TSF shall provide authorized users with the capability to verify the integrity of stored TSF executable code.

5.1.42 - Limited fault tolerance (FRU_FLT.2)

FRU_FLT.2.1 - The TSF shall ensure the operation of all the TOE's capabilities when [*ST assignment: list of type of failures*] occur.

5.1.43 - Basic limitation on multiple concurrent sessions (FTA_MCS.1)

FTA_MCS.1.1 - The TSF shall restrict the maximum number of concurrent sessions that belong to the same user.

FTA_MCS.1.2 - The TSF shall enforce, by default, a limit of [*one*] session per user.

5.1.44 - TSF-initiated termination (FTA_SSL.3)

FTA_SSL.3.1 - The TSF shall terminate an interactive session after a [*ST assignment: time interval of user inactivity*].

5.1.45 - TOE access history (FTA_TAH.1)

FTA_TAH.1.1 - Upon successful session establishment, the TSF shall display the [*date, time*] of the last successful session establishment to the user.

FTA_TAH.1.2 - Upon successful session establishment, the TSF shall display the [*date, time*] of the last unsuccessful attempt to session establishment and the number of unsuccessful attempts since the last successful session establishment.

FTA_TAH.1.3 - The TSF shall not erase the access history information from the user interface without giving the user an opportunity to review the information.

5.1.46 - TOE session establishment (FTA_TSE.1)

FTA_TSE.1.1 - The TSF shall be able to deny session establishment based on [*ST assignment: attributes*].

5.2 - Security Functional Requirements for the Environment

5.2.1 – Selectable Audit Review (FAU_SAR.3)

FAU_SAR.3.1 The TSF shall provide the ability to perform [*selection: searches, sorting, ordering*] of audit data based on [*assignment: criteria with logical relations*].

5.2.2 - Guarantees of audit data availability (FAU_STG.2)

FAU_STG.2.1 - The TSF shall protect the stored audit records from unauthorized deletion.

FAU_STG.2.2 - The TSF shall be able to [*detect and prevent*] modifications to the audit records.

FAU_STG.2.3 - The TSF shall ensure that [*ST assignment: time and size metric for saving audit records*] audit records will be maintained when the following conditions occur: [*audit storage exhaustion, failure, or attack*].

Note: The metric for saving audit records may be expressed as most recent number of days or most recent number of records. Records may be segregated based on the priority of service.

5.2.3 - Action in case of possible audit data loss (FAU_STG.3)

FAU_STG.3.1 - The TSF shall take [*ST assignment: actions to be taken in case of possible audit storage failure*] if the audit trail exceeds [*ST assignment: pre-defined limit*].

5.2.4 - Prevention of audit data loss (FAU_STG.4)

FAU_STG.4.1 - The TSF shall [*ST selection: `ignore auditable events', `prevent auditable events, except those taken by the authorized user with special rights', `overwrite the oldest stored audit records'*] and [*ST assignment: other actions to be taken in case of audit storage failure*] if the audit trail is full.

5.2.5 – Verification of Secrets (FIA_SOS.1)

FIA_SOS.1.1 - The TSF shall provide a mechanism to verify that secrets meet [*assignment: a defined quality metric]*.

5.2.6 - Management of security functions behavior (FMT_MOF.1)

FMT_MOF.1.1 - The TSF shall restrict the ability to [ST selection: determine the behavior of, disable, enable, modify the behavior of] the functions [ST assignment: list of functions] to [audit administrator, administrator, maintainer, or user].

5.2.7 - Management of security attributes (FMT_MSA.1)

FMT_MSA.1.1 - The TSF shall enforce the [ST assignment: access control SFP, information flow control SFP] to restrict the ability to [change default, query, modify, delete][ST assignment: other operations] the security attributes [ST assignment: list of security attributes] to [administrator]**.**

5.2.8 - Management of limits on TSF data (FMT_MTD.2)

FMT_MTD.2.1 - The TSF shall restrict the specification of the limits for [*ST assignment: list of TSF data*] to [*administrator*].

FMT_MTD.2.2 - The TSF shall take the following actions, if the TSF data are at, or exceed, the indicated limits: [*ST assignment: actions to be taken*].

5.2.9 - Secure TSF data (FMT_MTD.3)

FMT_MTD.3.1 - The TSF shall ensure that only secure values are accepted for TSF data.

5.2.10 - Revocation (FMT_REV.1)

FMT_REV.1.1 - The TSF shall restrict the ability to revoke security attributes associated with the [*subjects, objects, administrator, audit administrator, maintainer, and user roles*] within the TSC to [*administrator*].

FMT_REV.1.2 - The TSF shall enforce the rules [*ST assignment: specification of revocation rules*].

5.2.11 - Time-limited authorization (FMT_SAE.1)

FMT_SAE.1.1 - The TSF shall restrict the capability to specify an expiration time for [*ST assignment: list of security attributes for which expiration is to be supported, including the maintainer role*] to [*administrator*].

FMT_SAE.1.2 - For each of these security attributes, the TSF shall be able to [*assignment: list of actions to be taken for each security attribute*] after the expiration time for the indicated security attribute has passed.

5.2.12 - Restrictions on security roles (FMT_SMR.2)

FMT_SMR.2.1 - The TSF shall maintain the roles: [*the authorized identified roles of user and administrator*].

FMT_SMR.2.2 - The TSF shall be able to associate users with roles.

FMT_SMR.2.3 - The TSF shall ensure that the conditions [ST *assignment: conditions for the different roles*] are satisfied.

5.2.13 - User-initiated locking (FTA_SSL.2)

FTA_SSL.2.1 - The TSF shall allow user-initiated locking of the user's own interactive session, by:

a) clearing or overwriting display devices, making the current contents unreadable;

b) disabling any activity of the user's data access/display devices other than unlocking the session.

FTA_SSL.2.2 - The TSF shall require the following events to occur prior to unlocking the session: [*re-authentication at the TOE*]

5.2.14 - Default TOE access banners (FTA_TAB.1)

FTA_TAB.1.1 - Before establishing a user session, the TSF shall display an advisory warning message regarding unauthorized use of the TOE.

5.3 - TOE Security Assurance Requirements for the Environment

The TOE security assurance requirements described below detail the management and evaluative activities required to develop the CCS for use in the operational environment. For a justification of the security assurance requirements and the Evaluation Assurance Level selected, EAL 5, augmented, please reference Section 6.2.2.

5.3.1 - Configuration management (ACM)

5.3.1.1 - Complete CM automation (ACM_AUT.2)

ACM_AUT.2.1C - The CM system shall provide an automated means by which only authorized changes are made to the TOE implementation representation, and to all other configuration items.

ACM_AUT.2.1D - The developer shall use a CM system.

ACM_AUT.2.2C - The CM system shall provide an automated means to support the generation of the TOE.

ACM_AUT.2.2D - The developer shall provide a CM plan.

ACM_AUT.2.3C - The CM plan shall describe the automated tools used in the CM system.

ACM_AUT.2.4C - The CM plan shall describe how the automated tools are used in the CM system.

ACM_AUT.2.5C - The CM system shall provide an automated means to ascertain the changes between the TOE and its preceding version.

ACM_AUT.2.6C - The CM system shall provide an automated means to identify all other configuration items that are affected by the modification of a given configuration item.

5.3.1.2 - Advanced support (ACM_CAP.5)

ACM_CAP.5.1C - The reference for the TOE shall be unique to each version of the TOE.

ACM_CAP.5.1D - The developer shall provide a reference for the TOE.

ACM_CAP.5.2C - The TOE shall be labeled with its reference.

ACM_CAP.5.2D - The developer shall use a CM system.

ACM_CAP.5.3C - The CM documentation shall include a configuration list, a CM plan, an acceptance plan, and integration procedures.

ACM_CAP.5.3D - The developer shall provide CM documentation.

ACM_CAP.5.4C - The configuration list shall describe the configuration items that comprise the TOE.

ACM_CAP.5.5C - The CM documentation shall describe the method used to uniquely identify the configuration items.

ACM_CAP.5.6C - The CM system shall uniquely identify all configuration items.

ACM_CAP.5.7C - The CM plan shall describe how the CM system is used.

ACM_CAP.5.8C - The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.

ACM_CAP.5.9C - The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.

ACM_CAP.5.10C - The CM system shall provide measures such that only authorized changes are made to the configuration items.

ACM_CAP.5.11C - The CM system shall support the generation of the TOE.

ACM_CAP.5.12C - The acceptance plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.

ACM_CAP.5.13C - The integration procedures shall describe how the CM system is applied in the TOE manufacturing process.

ACM_CAP.5.14C - The CM system shall require that the person responsible for accepting a configuration item into CM is not the person who developed it.

ACM_CAP.5.15C - The CM system shall clearly identify the configuration items that comprise the TSF.

ACM_CAP.5.16C - The CM system shall support the audit of all modifications to the TOE, including as a minimum the originator, date, and time in the audit trail.

ACM_CAP.5.17C - The CM system shall be able to identify the master copy of all material used to generate the TOE.

ACM_CAP.5.18C - The CM documentation shall demonstrate that the use of the CM system, together with the development security measures, allow only authorized changes to be made to the TOE.

ACM_CAP.5.19C - The CM documentation shall demonstrate that the use of the integration procedures ensures that the generation of the TOE is correctly performed in an authorized manner.

ACM_CAP.5.20C - The CM documentation shall demonstrate that the CM system is sufficient to ensure that the person responsible for accepting a configuration item into CM is not the person who developed it.

ACM_CAP.5.21C - The CM documentation shall justify that the acceptance procedures provide for an adequate and appropriate review of changes to all configuration items.

5.3.1.3 - Development tools CM coverage (ACM_SCP.3)

ACM_SCP.3.1C - The CM documentation shall show that the CM system, as a minimum, tracks the following: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, security flaws, and development tools and related information.

ACM_SCP.3.1D - The developer shall provide CM documentation.

ACM_SCP.3.2C - The CM documentation shall describe how configuration items are tracked by the CM system.

5.3.2 - Delivery and operation (ADO)

5.3.2.1 - Prevention of modification (ADO_DEL.3)

ADO_DEL.3.1C - The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.

ADO_DEL.3.1D - The developer shall document procedures for delivery of the TOE or parts of it to the user.

ADO_DEL.3.2C - The delivery documentation shall describe how the various procedures and technical measures provide for the prevention of modifications, or any discrepancy between the developer's master copy and the version received at the user site.

ADO_DEL.3.2D - The developer shall use the delivery procedures.

ADO_DEL.3.3C - The delivery documentation shall describe how the various procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.

5.3.2.2 - Generation log (ADO_IGS.2)

ADO_IGS.2.1C - The documentation shall describe the steps necessary for secure installation, generation, and start-up of the TOE.

ADO_IGS.2.1D - The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.

ADO_IGS.2.2C - The documentation shall describe procedures capable of creating a log containing the generation options used to generate the TOE in such a way that it is possible to determine exactly how and when the TOE was generated.

5.3.3 - Development (ADV)

5.3.3.1 - Semiformal functional specification (ADV_FSP.3)

ADV_FSP.3.1C - The functional specification shall describe the TSF and its external interfaces using a semiformal style, supported by informal, explanatory text where appropriate.

ADV_FSP.3.1D - The developer shall provide a functional specification.

ADV_FSP.3.2C - The functional specification shall be internally consistent.

ADV_FSP.3.3C - The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing complete details of all effects, exceptions and error messages.

ADV_FSP.3.4C - The functional specification shall completely represent the TSF.

ADV_FSP.3.5C - The functional specification shall include rationale that the TSF is completely represented.

5.3.3.2 - Semiformal high-level explanation (ADV_HLD.4)

ADV_HLD.4.1C - The presentation of the high-level design shall be semiformal.

ADV_HLD.4.1D - The developer shall provide the high-level design of the TSF.

ADV_HLD.4.2C - The high-level design shall be internally consistent.

ADV_HLD.4.3C - The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV_HLD.4.4C - The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV_HLD.4.5C - The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV_HLD.4.6C - The high-level design shall identify all interfaces to the subsystems of the TSF.ADV_HLD.4.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

ADV_HLD.4.8C - The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing complete details of all effects, exceptions and error messages.

ADV_HLD.4.9C - The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.

ADV_HLD.4.10C - The high-level design shall justify that the identified means of achieving separation, including any protection mechanisms, are sufficient to ensure a clear and effective separation of TSP-enforcing from non-TSP-enforcing functions.

ADV_HLD.4.11C - The high-level design shall justify that the TSF mechanisms are sufficient to implement the security functions identified in the high-level design.

5.3.3.3 - Structured implementation of the TSF (ADV_IMP.3)

ADV_IMP.3.1C - The implementation representation shall unambiguously define the TSF to a level of detail such that the TSF can be generated without further design decisions.

ADV_IMP.3.1D - The developer shall provide the implementation representation for the entire TSF.

ADV_IMP.3.2C - The implementation representation shall be internally consistent.

ADV_IMP.3.3C - The implementation representation shall describe the relationships between all portions of the implementation.

ADV_IMP.3.4C The implementation representation shall be structured into small and comprehensible sections.

5.3.3.4 - Modularity (ADV_INT.1)

ADV_INT.1.1C The architectural description shall identify the modules of the TSF.

ADV_INT.1.1D The developer shall design and structure the TSF in a modular fashion that avoids unnecessary interactions between the modules of the design.

ADV_INT.1.2C The architectural description shall describe the purpose, interface, parameters, and effects of each module of the TSF.

ADV_INT.1.2D The developer shall provide an architectural description.

ADV_INT.1.3C The architectural description shall describe how the TSF design provides for largely independent modules that avoid unnecessary interactions.

5.3.3.5 - Semiformal low-level design (ADV_LLD.2)

ADV_LLD.2.1C - The presentation of the low-level design shall be semiformal.

ADV_LLD.2.1D - The developer shall provide the low-level design of the TSF.

ADV_LLD.2.2C - The low-level design shall be internally consistent.

ADV_LLD.2.3C - The low-level design shall describe the TSF in terms of modules.

ADV_LLD.2.4C - The low-level design shall describe the purpose of each module.

ADV_LLD.2.5C - The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.

ADV_LLD.2.6C - The low-level design shall describe how each TSP-enforcing function is provided.

ADV_LLD.2.7C - The low-level design shall identify all interfaces to the modules of the TSF.

ADV_LLD.2.8C - The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.

ADV_LLD.2.9C - The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing complete details of all effects, exceptions and error messages.

ADV_LLD.2.10C - The low-level design shall describe the separation of the TOE into TSP-enforcing and other modules.

5.3.3.6 - Semiformal correspondence demonstration (ADV_RCR.2)

ADV_RCR.2.1C - For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

ADV_RCR.2.1D - The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.

ADV_RCR.2.2C - For each adjacent pair of provided TSF representations, where portions of both representations are at least semiformally specified, the demonstration of correspondence between those portions of the representations shall be semiformal.

5.3.3.7 - Informal TOE security policy model (ADV_SPM.1)

ADV_SPM.1.1C - The TSP model shall be informal.

ADV_SPM.1.1D - The developer shall provide a TSP model.

ADV_SPM.1.2C - The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modeled.

ADV_SPM.1.2D - The developer shall demonstrate correspondence between the functional specification and the TSP model.

ADV_SPM.1.3C - The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modeled.

ADV_SPM.1.4C - The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.

5.3.4 - Guidance documents (AGD)

5.3.4.1 - Administrator guidance (AGD_ADM.1)

AGD_ADM.1.1C - The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.

AGD_ADM.1.1D - The developer shall provide administrator guidance addressed to system administrative personnel.

AGD_ADM.1.2C - The administrator guidance shall describe how to administer the TOE in a secure manner.

AGD_ADM.1.3C - The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD_ADM.1.4C - The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the TOE.

AGD_ADM.1.5C - The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD_ADM.1.6C - The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_ADM.1.7C - The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD_ADM.1.8C - The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

5.3.4.2 - User guidance (AGD_USR.1)

AGD_USR.1.1C - The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD_USR.1.1D - The developer shall provide user guidance.

AGD_USR.1.2C - The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD_USR.1.3C - The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD_USR.1.4C - The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behavior found in the statement of TOE security environment.

AGD_USR.1.5C - The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD_USR.1.6C - The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

5.3.5 - Life cycle support (ALC)

5.3.5.1 - Sufficiency of security measures (ALC_DVS.2)

ALC_DVS.2.1C - The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC_DVS.2.1D - The developer shall produce development security documentation.

ALC_DVS.2.2C - The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.

ALC_DVS.2.3C - The evidence shall justify that the security measures provide the necessary level of protection to maintain the confidentiality and integrity of the TOE.

5.5.5.2 - Systematic flaw remediation (ALC_FLR.3)

ALC_FLR.3.1C - The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.3.1D - The developer shall document the flaw remediation procedures.

ALC_FLR.3.2C - The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.3.2D - The developer shall establish a procedure for accepting and acting upon user reports of security flaws and requests for corrections to those flaws.

ALC_FLR.3.3C - The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.3.3D - The developer shall designate one or more specific points of contact for user reports and inquiries about security issues involving the TOE.

ALC_FLR.3.4C - The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

ALC_FLR.3.5C - The procedures for processing reported security flaws shall ensure that any reported flaws are corrected and the correction issued to TOE users.

ALC_FLR.3.6C - The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.

ALC_FLR.3.7C - The flaw remediation procedures shall include a procedure requiring timely responses for the automatic distribution of security flaw reports and the associated corrections to registered users who might be affected by the security flaw.

5.3.5.3 - Standardized life-cycle model (ALC_LCD.2)

ALC_LCD.2.1C - The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.

ALC_LCD.2.1D - The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

ALC_LCD.2.2C - The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.

ALC_LCD.2.2D - The developer shall provide life-cycle definition documentation.

ALC_LCD.2.3C - The life-cycle definition documentation shall explain why the model was chosen.

ALC_LCD.2.3D - The developer shall use a standardized life-cycle model to develop and maintain the TOE.

ALC_LCD.2.4C - The life-cycle definition documentation shall explain how the model is used to develop and maintain the TOE.

ALC_LCD.2.5C - The life-cycle definition documentation shall demonstrate compliance with the standardized life-cycle model.

5.3.5.4 - Compliance with implementation standards - all parts (ALC_TAT.3)

ALC_TAT.3.1C - All development tools used for implementation shall be well-defined.

ALC_TAT.3.1D - The developer shall identify the development tools being used for the TOE.

ALC_TAT.3.2C - The documentation of the development tools shall unambiguously define the meaning of all statements used in the implementation.

ALC_TAT.3.2D - The developer shall document the selected implementation-dependent options of the development tools.

ALC_TAT.3.3C - The documentation of the development tools shall unambiguously define the meaning of all implementation-dependent options.

ALC_TAT.3.3D - The developer shall describe the implementation standards for all parts of the TOE.

5.3.6 - Maintenance of assurance (AMA)

5.3.6.1 - Assurance maintenance plan (AMA_AMP.1)

AMA_AMP.1.1C - The AM Plan shall contain or reference a brief description of the TOE, including the security functionality it provides.

AMA_AMP.1.1D - The developer shall provide an AM Plan.

AMA_AMP.1.2C - The AM Plan shall identify the certified version of the TOE, and shall reference the evaluation results.

AMA_AMP.1.3C - The AM Plan shall reference the TOE component categorization report for the certified version of the TOE.

AMA_AMP.1.4C - The AM Plan shall define the scope of changes to the TOE that are covered by the plan.

AMA_AMP.1.5C - The AM Plan shall describe the TOE life-cycle, and shall identify the current plans for any new releases of the TOE, together with a brief description of any planned changes that are likely to have a significant security impact.

AMA_AMP.1.6C - The AM Plan shall describe the assurance maintenance cycle, stating and justifying the planned schedule of AM audits and the target date of the next re-evaluation of the TOE.

AMA_AMP.1.7C - The AM Plan shall identify the individual(s) who will assume the role of developer security analyst for the TOE.

AMA_AMP.1.8C - The AM Plan shall describe how the developer security analyst role will ensure that the procedures documented or referenced in the AM Plan are followed.

AMA_AMP.1.9C - The AM Plan shall describe how the developer security analyst role will ensure that all developer actions involved in the analysis of the security impact of changes affecting the TOE are performed correctly.

AMA_AMP.1.10C - The AM Plan shall justify why the identified developer security analyst(s) have sufficient familiarity with the security target, functional specification and (where appropriate) high-level design of the TOE, and with the evaluation results and all applicable assurance requirements for the certified version of the TOE.

AMA_AMP.1.11C - The AM Plan shall describe or reference the procedures to be applied to maintain the assurance in the TOE, which as a minimum shall include the procedures for configuration management, maintenance of assurance evidence, performance of the analysis of the security impact of changes affecting the TOE, and flaw remediation.

5.3.6.2 - TOE component categorization report (AMA_CAT.1)

AMA_CAT.1.1C - The TOE component categorization report shall categories each component of the TOE, identifiable in each TSF representation from the most abstract to the least abstract, according to its relevance to security; as a minimum, TOE components must be categorized as one of TSP-enforcing or non-TSP- enforcing.

AMA_CAT.1.1D - The developer shall provide a TOE component categorization report for the certified version of the TOE.

AMA_CAT.1.2C - The TOE component categorization report shall describe the categorization scheme used, so that it can be determined how to categories new components introduced into the TOE, and also when to re-categories existing TOE components following changes to the TOE or its security target.

AMA_CAT.1.3C - The TOE component categorization report shall identify any tools used in the development environment that, if modified, will have an impact on the assurance that the TOE satisfies its security target.

5.3.6.3 - Evidence of maintenance process (AMA_EVD.1)

AMA_EVD.1.1C - The AM documentation shall include a configuration list and a list of identified vulnerabilities in the TOE.

AMA_EVD.1.1D - The developer security analyst shall provide AM documentation for the current version of the TOE.

AMA_EVD.1.2C - The configuration list shall describe the configuration items that comprise the current version of the TOE.

AMA_EVD.1.3C - The AM documentation shall provide evidence that the procedures documented or referenced in the AM Plan are being followed.

AMA_EVD.1.4C - The list of identified vulnerabilities in the current version of the TOE shall show, for each vulnerability, that the vulnerability cannot be exploited in the intended environment for the TOE.

5.3.6.4 - Examination of security impact analysis (AMA_SIA.2)

AMA_SIA.2.1C - The security impact analysis shall identify the certified TOE from which the current version of the TOE was derived.

AMA_SIA.2.1D - The developer security analyst shall, for the current version of the TOE, provide a security impact analysis that covers all changes affecting the TOE as compared with the certified version.

AMA_SIA.2.2C - The security impact analysis shall identify all new and modified TOE components that are categorized as TSP-enforcing.

AMA_SIA.2.3C - The security impact analysis shall, for each change affecting the security target or TSF representations, briefly describe the change and any effects it has on lower representation levels.

AMA_SIA.2.4C - The security impact analysis shall, for each change affecting the security target or TSF representations, identify all IT security functions and all TOE components categorized as TSP-enforcing that are affected by the change.

AMA_SIA.2.5C - The security impact analysis shall, for each change which results in a modification of the implementation representation of the TSF or the IT environment, identify the test evidence that shows, to the required level of assurance, that the TSF continues to be correctly implemented following the change.

AMA_SIA.2.6C - The security impact analysis shall, for each applicable assurance requirement in the configuration management (ACM), life cycle support (ALC), delivery and operation (ADO) and guidance documents (AGD) assurance classes, identify any evaluation deliverables that have changed, and provide a brief description of each change and its impact on assurance.

AMA_SIA.2.7C - The security impact analysis shall, for each applicable assurance requirement in the vulnerability assessment (AVA) assurance class, identify which evaluation deliverables have changed and which have not, and give reasons for the decision taken as to whether or not to update the deliverable.

5.3.7 - Tests (ATE)

5.3.7.1 - Rigorous analysis of coverage (ATE_COV.3)

ATE_COV.3.1C - The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

ATE_COV.3.1D - The developer shall provide an analysis of the test coverage.

ATE_COV.3.2C - The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.

ATE_COV.3.3C - The analysis of the test coverage shall rigorously demonstrate that all external interfaces of the TSF identified in the functional specification have been completely tested.

5.3.7.2 - Testing: implementation representation (ATE_DPT.3)

ATE_DPT.3.1C - The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design, low-level design and implementation representation.

ATE_DPT.3.1D - The developer shall provide the analysis of the depth of testing.

5.3.7.3 - Ordered functional testing (ATE_FUN.2)

ATE_FUN.2.1C - The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

ATE_FUN.2.1D - The developer shall test the TSF and document the results.

ATE_FUN.2.2C - The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE_FUN.2.2D - The developer shall provide test documentation.

ATE_FUN.2.3C - The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.2.4C - The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.2.5C - The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

ATE_FUN.2.6C - The test documentation shall include an analysis of the test procedure ordering dependencies.

5.3.7.4 - Independent testing - sample (ATE_IND.2)

ATE_IND.2.1C - The TOE shall be suitable for testing.

ATE_IND.2.1D - The developer shall provide the TOE for testing.

ATE_IND.2.2C - The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

5.3.8 - Vulnerability assessment (AVA)

5.3.8.1 - Exhaustive covert channel analysis (AVA_CCA.3)

AVA_CCA.3.1C - The analysis documentation shall identify covert channels and estimate their capacity.

AVA_CCA.3.1D - The developer shall conduct a search for covert channels for each information flow control policy.

AVA_CCA.3.2C - The analysis documentation shall describe the procedures used for determining the existence of covert channels, and the information needed to carry out the covert channel analysis.

AVA_CCA.3.2D - The developer shall provide covert channel analysis documentation.

AVA_CCA.3.3C - The analysis documentation shall describe all assumptions made during the covert channel analysis.

AVA_CCA.3.4C - The analysis documentation shall describe the method used for estimating channel capacity, based on worst case scenarios.

AVA_CCA.3.5C - The analysis documentation shall describe the worst case exploitation scenario for each identified covert channel.

AVA_CCA.3.6C - The analysis documentation shall provide evidence that the method used to identify covert channels is exhaustive.

5.3.8.2 - Analysis and testing for insecure states (AVA_MSU.3)

AVA_MSU.3.1C - The guidance documentation shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AVA_MSU.3.1D - The developer shall provide guidance documentation.

AVA_MSU.3.2C - The guidance documentation shall be complete, clear, consistent and reasonable.

AVA_MSU.3.2D - The developer shall document an analysis of the guidance documentation.

AVA_MSU.3.3C - The guidance documentation shall list all assumptions about the intended environment.

AVA_MSU.3.4C - The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).

AVA_MSU.3.5C - The analysis documentation shall demonstrate that the guidance documentation is complete.

5.3.8.3 - Strength of TOE security function evaluation (AVA_SOF.1)

AVA_SOF.1.1C - For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.

AVA_SOF.1.1D - The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.

AVA_SOF.1.2C - For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.

5.3.8.4 - Highly resistant (AVA_VLA.4)

AVA_VLA.4.1C - The documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

AVA_VLA.4.1D - The developer shall perform and document an analysis of the TOE deliverables searching for ways in which a user can violate the TSP.

AVA_VLA.4.2C - The documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.

AVA_VLA.4.2D - The developer shall document the disposition of identified vulnerabilities.

AVA_VLA.4.3C - The evidence shall show that the search for vulnerabilities is systematic.

AVA_VLA.4.4C - The analysis documentation shall provide a justification that the analysis completely addresses the TOE deliverables.

6 - Rationale

This chapter presents the mapping to show complete coverage of the Policies, Threats, and Assumptions by the Security Objectives (summarized in Table 6-1) and complete coverage of the Security Objective by the functional and assurance requirements.

6.1 - Security Objectives Rationale

Table 1 - Mapping the TOE Security Environment to Security Objectives

Policy/Threat	Assumption	Objectives	Environmental Objectives
P.ACCOUNT	A.TRUSTED_ADMIN, A.Trusted_User	O.AUDIT, O.Identify	OE.Audit_Maintain
P.AUTHORITIES	A.Operating_Sys, A.Trusted_Admin, A.TRUSTED_USER	O.Audit, O.Identify, O.Integrity	OE.ADMIN, OE.Audit_Maintain
P.AUTHORIZED_USE	A.Peer, A.Trusted_Admin, A.Trusted_User	O.Manage	OE.Admin, OE.Banner
P.AVAILABILITY	A.Operating_Sys, A.Trusted_Admin, A.Trusted_User	O.Integrity, O.Manage	OE.Admin, OE.Malicious_Code
P.CONFIDENTIALITY	A.Crypto, A.Peer A.Trusted_Admin, A.Trusted_User	O.CRYPTO_OPNS, O.Data_Exchange_Conf, O.MANAGE, O.No_Residual_Info	OE.ADMIN, OE.CRYPTO_DESIGN, OE.Malicious_Code, OE.OPERATE
P.GUIDANCE	A.Trusted_Admin, A.Trusted_User		OE.Operate
P.INFORMATION_AC	A.Operating_Sys, A.Trusted_Admin, A.Trusted_User	O.ACCESS, O.IDENTIFY, O.Manage	OE.Admin
P.INTEG	A.Crypto, A.Operating_Sys	O.CRYPTO_OPNS, O.Fault_Tolerant, O.Integrity	OE.Malicious_Code
P.LIFECYCLE	A.Trusted_Admin		OE.Admin, OE.CRYPTO_DESIGN, OE.OPERATE
P.MANAGE	A.Operating_Sys, A.Peer, A.Trusted_Admin	O.Integrity_Attr_Exch, O.Manage, O.Session_Termination	OE.Admin, OE.OPERATE
P.PHYSICAL_CONTROL	A.Trusted_Admin, A.Trusted_User	O.Integrity	
P.TEMPEST	A.Trusted_Admin, A.Trusted_User, A.Tempest		

Policy/Threat	Assumption	Objectives	Environmental Objectives
T.ALTER	A.Operating_Sys, A.Trusted_Admin, A.Trusted_User	O.ACCESS, O.Audit, O.Integrity, O.Manage	OE.Admin, OE.AUDIT_MAINTAIN, OE.OPERATE
T.COMPONENT_FAILURE	A.Operating_Sys	O.CRYPTO_OPNS, O.Fault_Tolerant, O.Integrity	OE.CRYPTO_DESIGN
T.CRASH	A.Operating_Sys, A.Peer	O.Fault_Tolerant	
T.DEV_FLAWED_CODE	A.Trusted_Admin		OE.ADMIN
T.ERROR	A.Crypto, A.Peer, A.Trusted_Admin	O.Audit, O.CRYPTO_OPNS, O.Identify	OE.Admin, OE.Audit_Maintain, OE.Operate
T.HACK_AC	A.Trusted_Admin	O.ACCESS, O.No_Residual_Info	OE.Admin
T.HACK_CRYPTO	A.Crypto	O.CRYPTO_OPNS	OE.CRYPTO_DESIGN
T.HACK_MASQ	A.Operating_Sys, A.Trusted_Admin, A.Trusted_User	O.MANAGE, O.No_Residual_Info, O.Session_Termination	
T.HACK_TRAFFIC	A.Crypto	O.CRYPTO_OPNS	OE.CRYPTO_DESIGN
T.IMPORT	A.Peer, A.Trusted_Admin, A.Trusted_User	O.Integrity	OE.Admin, OE.Malicious_Code, OE.OPERATE
T.PHYSICAL	A.Operating_Sys, A.Trusted_Admin, A.Trusted_User	O.Integrity	

Table 2 - Tracing of Security Objectives to the TOE Security Environment

Security Objectives for the TOE	
Objective	**Policies/Threats/Assumptions**
O.ACCESS	P.CONFIDENTIALITY, P.Information_AC, T.ALTER, T.Hack_AC
O.AUDIT	A.OPERATING_SYS, A.TRUSTED_ADMIN, A.TRUSTED_USER, P.ACCOUNT, P. AUTHORITIES, T.ALTER, T.ERROR
O.CRYPTO_OPNS	A.CRYPTO, P.CONFIDENTIALITY, P.INTEG, T.Component_Failure, T.ERROR, T.Hack_Crypto
O.Data_Exchange_Conf	P.CONFIDENTIALITY
O.FAULT_TOLERANT	A.OPERATING_SYS, P.INTEG, T.Component_Failure, T.CRASH,
O.IDENTIFY	P.ACCOUNT, P.AUTHORITIES, P.CONFIDENTIALITY, P.Information_AC, T.ERROR
O.INTEGRITY	A.OPERATING_SYS, A.TRUSTED_ADMIN, P.AUTHORITIES, P.Availability, P.INTEG, P.Physical_Control, T.ALTER, T.Component_Failure, T.IMPORT, T.PHYSICAL
O.Integrity_Attr_Exch	P.MANAGE
O.MANAGE	A.TRUSTED_ADMIN, P.AUTHORIZED_USE, P.AVAILABILITY, P.CONFIDENTIALITY, P.Information_AC, P.MANAGE, T.ALTER, T.Hack_Masq
O.No_Residual_Info	P.CONFIDENTIALITY, T.Hack_AC, T.Hack_Masq
O.Session_Termination	A.OPERATING_SYS, P.MANAGE, T.Hack_Masq

Table 3 - Tracing of Security Objectives to the TOE Security Environment

Security Objectives for the Environment	
Environmental Objective	**Policies/Threats/Assumptions**
OE.ADMIN	A.OPERATING_SYS, A.Peer, A.TRUSTED_ADMIN, P.AUTHORITIES, P.Authorized_Use, P.Availability, P.CONFIDENTIALITY, P.Guidance, P.Information_AC, P.INTEG, P.Lifecycle, P.MANAGE, P.Physical_Control, T.ALTER, T.Component_Failure, T.CRASH, T.Dev_Flawed_Code, T.ERROR, T.Hack_AC, T.IMPORT
OE.AUDIT_MAINTAIN	A.OPERATING_SYS, A.TRUSTED_ADMIN, P.ACCOUNT, P.AUTHORITIES, P.Information_AC, P.MANAGE, P.Physical_Control, T.ALTER, T.ERROR
OE.BANNER	P.Authorized_Use
OE.CRYPTO_DESIGN	A.CRYPTO, P.CONFIDENTIALITY, P.Lifecycle, T.Component_Failure, T.Hack_Crypto
OE.Malicious_Code	P.Availability, P.CONFIDENTIALITY, P.INTEG, T.IMPORT
OE.OPERATE	A.OPERATING_SYS, A.Peer, A.TRUSTED_ADMIN, A.TRUSTED_USER, P.Authorized_Use, P.CONFIDENTIALITY, P.Guidance, P.Lifecycle, P.Physical_Control, T.ALTER, T.CRASH, T.ERROR, T.Hack_Masq, T.IMPORT
OE.Screen_Lock	T.Hack_Masq
OE.Source_Code_Exam	A.TRUSTED_ADMIN, T.Dev_Flawed_Code

6.1.1 - Policies

P.ACCOUNT: User activity shall be monitored so that they may be held accountable for their actions, sanctions can be applied when malfeasance occurs, and proper application of system controls is ensured. All users will be notified that such monitoring may occur.

Coverage Rationale: To ensure that the user is held accountable, O.AUDIT records specified actions done by the user. O.IDENTIFY will identify and authenticate all users before any action within the TOE is allowed. The environmental objective OE.AUDIT_MAINTAIN will ensure that the proper audit capabilities are functioning properly for accurate accountability. A.TRUSTED_USER assumes that the user is trusted. A.TRUSTED_ADMIN assumes that administrators are trusted and will review audit logs to help enforce accountability policies.

P.AUTHORITIES: Appropriate authorities shall be immediately notified of any threats or vulnerabilities affecting systems that process their data.

Coverage Rationale: Through systematic security checks, O.INTEGRITY, the TOE will detect and prevent unauthorized changes to the system configuration. O.IDENTIFY, O.AUDIT, and OE.AUDIT_MAINTAIN will reveal to the administrator both authorized and unauthorized actions attributed to both authorized and unauthorized TOE. A.TRUSTED_ADMIN assumes that the system administrator follows policies and procedures regarding the secure administration of the TOE. OE.ADMIN and OE.AUDIT_MAINTAIN require administrators to verify integrity data, review audit logs, and notify authorities of any misuse or tampering of the system. A.OPERATING_SYS assumes that the system is functioning properly and that operating system vulnerabilities cannot be used to subvert trust in the audit or integrity data. A.TRUSTED_USER assumes that all users of the system are trusted.

P.AUTHORIZED USE: Information shall be used only for its authorized purpose(s).

Coverage Rationale: O.MANAGE will provide features to enhance security of the TOE through limiting the roles and identities of authorized users of the TOE, so, while there may be many users of the TOE, there will be relatively few administrators. OE.ADMIN enables the administrators to manage authorizations, system security, and other security relevant data within the TOE. The system will provide a banner notifying users of the security of the system and monitoring of their actions, OE.BANNER. To maintain a secure level of operation, A.TRUSTED_ADMIN assumes that the administrators are informed of the proper use of the TOE and the information thereon. A.PEER also assumes that the systems with which the TOE communicates operates under the same secure policies. A.TRUSTED_ADMIN and A.TRUSTED_USER both assume that the users and administrators are trusted and authorized to use the TOE.

P.AVAILABILITY: Information shall be available to satisfy mission requirements.

Coverage Rationale: O.INTEGRITY will ensure that the system security information is accurate through periodic checks of its security functions. OE.ADMIN enables

administrators to manage, through O.MANAGE, access privileges to information to mission personnel. OE.MALICIOUS_CODE provides security to the TOE by allowing system administrators to incorporate malicious code prevention procedures into the system. A.TRUSTED_ADMIN assumes that the administrators follow procedures according to the TOE to allow data availability to authorized users. A.OPERATING_SYS assumes that the operating system functions properly to have information readily available. A.TRUSTED_USER assumes users are trusted to use available information to achieve the mission.

P.CONFIDENTIALITY: The confidentiality of user and system data stored or processed in the TOE must be protected.

Coverage Rationale: O.CRYPTO_OPNS will ensure that all data passed will be encrypted adequately to prevent unauthorized use of the information. O.Data_Exchange_Conf will ensure the user data is protected as well as the information being passed. O.No_Residual_Info will ensure that data previously used cannot be accessed again once the system resource is reassigned. OE.CRYPTO_DESIGN provides requirements for developers to implement cryptographic designs within the TOE. A.TRUSTED_ADMIN assumes that the administrators will operate systems in a competent and confidential manner. Administrators, through OE.ADMIN and O.MANAGE, guided by OE.OPERATE, limit help ensure data confidentiality through privileges to authorized users. O.ACCESS and O.IDENTIFY ensure that data is shared only with properly identified people. OE.MALICIOUS_CODE provides security to the TOE by allowing system administrators to incorporate malicious code prevention procedures into the system. A.CRYPTO assumes that the cryptographic design and operations are suitable to the data and threat. A.PEER assumes that any system communicating with the TOE will operate at the same security level and under the same management procedures as the TOE. A.TRUSTED_USER further assumes that the users are trusted and will operate the system with appropriate confidentiality.

P.GUIDANCE: Guidance shall be provided for the secure installation, administration, and use of the system.

Coverage Rationale: OE.OPERATE provides system administrators and users policies and procedures for secure installation, use, and administration of the TOE. A.TRUSTED_ADMIN and A.TRUSTED_USER assume that the users and administrators follow the guidance.

P.INFORMATION_AC: Only authorized individuals and processes shall access information.

Coverage Rationale: O.MANAGE maintains privileges of users and administrators. Administrators set privileges through OE.ADMIN. A.TRUSTED_ADMIN assumes that system administrators can and will follow procedures to manage information access. A.OPERATING_SYS assumes that the operating system functions properly to not override or undermine TOE information access controls. O.ACCESS grants access according the privileges set in O.MANAGE based on the identity, established in O.IDENTITY, of the authorized user. A.TRUSTED_USER assumes that the user of the system is trustworthy to ensure that data and processing remain secure.

P.INTEG: The integrity of user and system data stored or processed in the TOE must be protected.

Coverage Rationale: O.CRYPTO_OPNS will protect the TOE by using a cryptographic hash on the audit trail and encrypting external communications. The cryptographic module will protect itself through encryption of its own data and self-tests. O.FAULT_TOLERANT ensures that any information is protected in the event the TOE is partially incapacitated. O.INTEGRITY will ensure that the system information is secure through its periodic checks of its security functions. OE.Malicious_Code secures the data by integrating malicious code prevention procedures to protect the TOE data. A.CRYPTO assumes that the cryptographic operations and design are suitable to the data and threat environment. A.OPERATING_SYS assumes that the operating system functions properly to not override or undermine TOE integrity functions.

P.LIFECYCLE: Information systems security shall be an integral part of all system lifecycle phases.

Coverage Rationale: OE.OPERATE and OE.CRYPTO_DESIGN provide development guidance and operational requirements. In OE.ADMIN and A.TRUSTED_ADMIN, trusted administrators manage the security of TOE throughout the lifecycle of the system.

P.MANAGE: The TOE shall be managed such that its security functions are implemented and preserved throughout its operational lifetime.

Coverage Rationale: O.Integrity_Attr_Exch ensures the system safely exchanges information with another trusted IT product. A.PEER ensures that the TOE only communicates with systems that are managed in the same manner. OE.ADMIN requires administrators to operate the TOE in a secure manner. O.MANAGE will provide features to aid in managing the security functions of the TOE properly. O.SESSION_TERMINATION aids in managing a secure system by terminating the system after a determined period of inactivity. Guidelines to administrators will be provided through OE.OPERATE documents. It is assumed that administrators can and will follow guidelines to improve secure administration of the system as well as be trusted to ensure the security of the system in A.TRUSTED_ADMIN.

P.PHYSICAL_CONTROL: Information shall be physically protected to prevent unauthorized disclosure, destruction, or modification.

Coverage Rationale: O.INTEGRITY requires that the TOE protect itself from tampering. A.TRUSTED_ADMIN and A.TRUSTED_USER assume that administrators and users are physically capable and trustworthy enough to protect the system.

P.TEMPEST: The TOE shall be constructed such that all emanations of red data satisfy the customer TEMPEST requirements.

Coverage Rationale: Evaluation of electromagnetic emanation requirements are explicitly excluded from the Common Criteria (ISO 15408). ISO 15408 recommends that organizational security policy statements be used to define the emanation controls required for the TOE. Secure Usage Assumptions should be used to articulate the requirement for the TOE to implement that policy. Assumptions should also be used to

specify any procedural and physical measures that need to be taken to prevent the detection of electromagnetic emanations by unauthorized individuals or users, or to prevent unwanted electromagnetic radiation. Thus, the assumption A.TEMPEST, supported by A.TRUSTED_ADMIN and A.TRUSTED_USER (to assume that neither disables tempest protection measures or uses the TOE in a situation where TEMPEST measures are ineffective) help support this policy.

6.1.2 - Threats

T.ALTER: An unauthorized user may surreptitiously gain access to the TOE and attempt to alter, replace, and/or deny access to system elements (e.g. hardware, firmware, or software) in an attempt to subvert the device.

Coverage Rationale: O.ACCESS will limit the actions of a user to the TOE in the event that too many attempts to access the system are attempted. O.AUDIT and OE.AUDIT_MAINTAIN will log, store, and protect the data trail of everyone who accesses protected processes and data in the TOE. O.INTEGRITY will ensure that critical information will be intact and all security attributes remain secure for the protection of the TSF data. O.MANAGE controls and specifies what users can do on particular objects. It is assumed that administrators follow procedures to prevent a security breach (A.TRUSTED_ADMIN). In OE.ADMIN, administrators will take steps to stop anomalous activities identified in the audit trail. A.OPERATING_SYS assumes that the operating system functions properly to not override or undermine TOE integrity functions. Both the administrators and users are trusted to ensure the security of the TOE (A.TRUSTED_ADMIN, A.TRUSTED_USER).

T.COMPONENT_FAILURE: Failure of one or more system components results in the loss of system-critical functionality.

Coverage Rationale: O.CRYPTO_OPNS will maintain encryption code security in the event of system failure. O.FAULT_TOLERANT ensures that the TOE will be able to perform mission critical functions even in the event of a crash. O.INTEGRITY will ensure that critical information will be intact and all security attributes remain in effect for the protection of the TSF data. A well-designed cryptographic module (HCT) within the TOE should not introduce weaknesses that would contribute to supporting T.COMPONENT_FAILURE (OE.CRYPTO_DESIGN). A.OPERATING_SYS assumes that the data remains secure in the event of an operating system failure.

T.CRASH: Due to interruption of the operation of the TOE resulting from power failure or other unforeseen interruptions, security critical information is either incomplete or corrupted.

Coverage Rationale: O.FAULT_TOLERANT ensures that the TOE will be able to perform mission critical functions even in the event of a crash. A.OPERATING_SYS assumes that the data remains secure in the event of an operating system failure from such a root cause. A.PEER assumes that any system with which the TOE communicates will operate under the same strict security regulations so that a crash in the companion system does not adversely affect the TOE.

T.DEV_FLAWED_CODE: A system or applications developer delivers code that does not perform according to specifications or contains security flaws.

Coverage Rationale: OE.ADMIN requires that when known vulnerabilities are discovered in code on the TOE, the administrator will apply code fixes after they are evaluated and that the traversing the TOE to the remote host or secure enclave is authenticated and authorized before any sent software execution is done. System administrators are assumed to be competent and trustworthy to initiate codes into the system (A.TRUSTED_ADMIN).

T.ERROR: An authorized user or administrator may perform erroneous actions that will compromise user and/or system resources.

Coverage Rationale: O.AUDIT will aid in determining the level of damage caused by the error. OE.AUDIT_MAINTAIN will provide administrators the documentation in providing security for the audit facilities and audit storage. The audit trail is then used as a training tool to prevent future errors. A.PEER and O.CRYPTO_OPNS will prevent data from being sent to an improperly managed site, and if the data is sent anyway, encryption protects it from unauthorized use. O.IDENTIFY will control access to the TOE and identify people or systems for remedial training or patching. OE.ADMIN, A.TRUSTED_ADMIN, and OE.OPERATE will provide guidance to the administrators trusted to limit the propagation of errors in the TOE. A.CRYPTO assumes that proper encryption/decryption methods have been provided for the cryptographic module. It is assumed that the system with which the TOE communicates with will managed under the same security rules (A.PEER).

T.HACK_AC: A hacker gains undetected access to a system due to missing, weak and/or incorrectly implemented access control causing potential violations of integrity, confidentiality, or availability.

Coverage Rationale: O.ACCESS will help mitigate this threat by allowing proper access through identification and accountability. O.No_Residual_Info will ensure that no previously used information can be accessed and used when TOE operation is interrupted, incapacitated, or relocated. OE.ADMIN will ensure that the administrator follows the guidelines set to heighten security of the TOE. It is assumed that system administrators can follow procedures to increase the security of the TOE (A.TRUSTED_ADMIN).

T.HACK_CRYPTO: A hacker performs cryptanalysis on encrypted data in order to recover message content.

Coverage Rationale: O.CRYPTO_OPNS and OE.CRYPTO_DESIGN ensure that strong cryptographic algorithms are used to protect the system. A.CRYPTO assumes that the cryptologic design and implementation are appropriate under the circumstances of use.

T.HACK_MASQ: A hacker masquerades as an authorized user to perform operations that will be attributed to the authorized user or a system process.

Coverage Rationale: O.MANAGE limits the authority of any authorized user to conduct actions on the TOE and prevents authorized users from having more than one active

session open simultaneously. O.No_Residual_Info will prevent authorized user data from being replayed by anyone other than the authorized user. O.Session_Termination will terminate a session after a specified period of inactivity. A.TRUSTED_ADMIN and A.TRUSTED_USER assume that administrators and users of the TOE follow defined guidelines to prevent unauthorized access of the TOE. It is also assumed that the operating system functions correctly to prevent hacker access (A.OPERATING_SYS).

T.HACK_TRAFFIC: A hacker or an eavesdropper performs traffic analysis on message traffic to gather intelligence (e.g. indicators or warnings of the intentions of the person or organization sending or receiving messages).

Coverage Rationale: O.CRYPTO_OPNS and OE.CRYPTO_DESIGN ensure that cryptographic algorithms are used to protect message metadata. A.CRYPTO assumes that the cryptologic design and implementation are appropriate under the circumstances of use.

T.IMPORT: An authorized user, administrator, and/or remote IT system of the TOE may unwittingly introduce malicious code into the system, resulting in a compromise of the integrity, availability, and/or confidentiality of user and/or system resources.

Coverage Rationale: O.INTEGRITY will protect the system by detecting and preventing any tampering by users through periodic checks on both the system and user data. OE.OPERATE provides guidelines for administrators and users to use the system. OE.ADMIN will ensure that the code being sent is authenticated and authorized before any sent software execution is done. OE.MALICIOUS_CODE will protect the TOE by allowing administrators to provide malicious code prevention procedures to the system. A.TRUSTED_ADMIN and A.TRUSTED_USER assume that administrators and users follow defined policies and procedures to minimize the risk of importing malicious code. In order to prevent malicious data entering the system, it is assumed that any system communicating with the TOE is managed the same way as the TOE (A.PEER).

T.PHYSICAL: Security-critical parts of the TOE may be subject to physical attack by agents to compromise security.

Coverage Rationale: O.INTEGRITY will enhance the security of the system by preventing tampering. A.TRUSTED_ADMIN and A.TRUSTED_USER assume that the administrators and users will not put the TOE in unnecessarily physically risky situations. A.OPERATING_SYS assumes that the operating system will not undermine other protective measures taken to prevent physical threats to the TOE.

6.2 - Security Requirements Rationale

6.2.1 - Functional Security Requirements Rationale

6.2.1.1 – TOE Objectives Functional Security Requirements Rationale

The following tables map the functional and assurance requirements to the TOE Security Objectives and to the Environmental Objectives. After the tables, the PP authors provide rationale to show that the security and assurance requirements are suitable to meeting the objectives.

Table 4 - Functional Component to Security Objective Mapping

Security Objective	Requirements
O.ACCESS	FTA_MCS.1, FTA_TSE.1, FIA_UAU.2
O.AUDIT	FAU_GEN.1, FAU_GEN.2, FAU_SEL.1, FIA_UID.2, FMT_MTD.1, FMT_SMR.2, FPT_STM.1, FTA_TAB.1, FTA_TAH.1
O.CRYPTO_OPNS	FCS_CKM.1, FCS_CKM.2, FCS_CKM.3, FCS_CKM.4, FCS_COP.1, FDP_ACC.1, FDP_ACF.1 , FDP_IFC.1, FDP_IFF.1, FDP_IFF.5, FDP_SDI.1, FMT_MTD.1, FMT_SMR.2, FPT_SEP.3, FPT_TST.1
O.Data_Exchange_Conf	FDP_ETC.1, FDP_ITC.1, FDP_UCT.1
O.FAULT_TOLERANT	FPT_FLS.1, FPT_RCV.1, FPT_RCV.4, FRU_FLT.2
O.IDENTIFY	FIA_AFL.1, FIA_ATD.1, FIA_UAU.2, FIA_UAU.7, FIA_UID.2, FIA_USB.1, FPT_STM.1, FTA_TAB.1
O.INTEGRITY	FDP_SDI.1, FDP_UIT.1, FMT_MSA.2, FPT_AMT.1, FPT_PHP.1, FPT_PHP.3, FPT_RVM.1, FPT_SEP.3, FPT_TST.1
O.Integrity_Attr_Exch	FPT_TDC.1
O.MANAGE	FDP_ACC.1, FDP_ACF.1, FIA_ATD.1, FIA_USB.1, FMT_MSA.3, FMT_SMR.2, FMT_SMR.3
O.No_Residual_Info	FDP_RIP.2
O.Session_Termination	FTA_SSL.3

O.ACCESS: The TOE will control access to information that is subject to the TOE security policy, based on the identity of the individuals, such that this policy cannot be bypassed in the TOE. The TOE will restrict the actions a user may perform before the TOE verifies the identity of the user and will provide mechanisms to limit the number of user initiated sessions open at one time.

Coverage Rationale: FTA_MCS.1 prohibits multiple sessions, thereby limiting unauthorized access to the system under the identity of an authorized user with a session already open. The TOE will prohibit access to the system based on attributes described

in FTA_TSE.1. FIA_UAU.2 provides that the system will require users to be authenticated before access is allowed.

O.AUDIT: The TOE will provide support for an audit trail to ensure each authenticated user and TOE administrator can be held accountable for his or her actions in the TOE. The audit trail will be of sufficient detail to reconstruct events in determining the cause or magnitude of compromise should a security violation or malfunction occur. The TOE will support the collection of an audit trail. It will not maintain that trail nor will it perform queries on the trail to generate reports. It may be possible to configure the TOE to report varying levels of audit data based on the audit policy in effect for a particular user. Those varying levels may also be adjusted based on the time of day, day of the week, duration of a trip or mission, etc. The audit function will display to the authenticated user the most recent successful and unsuccessful attempts to establish a session as the user. The TOE will deter modification or destruction of audit data through the creation of an audit administrator role. The TOE will use a cryptographic hash on the audit data to detect and deter tampering. The audit log will uniquely identify each user and record the date and time of action, action, the subject performing the action, and the object acted upon.

Coverage Rationale: FAU_GEN.1 describes the level of audit and the amount of information associated with auditable events that are recorded in the audit log. FAU_GEN.2 links users to auditable events, holding them accountable for their actions. PP/ST authors will determine selected information that is auditable based on their attributes according to FAU_SEL.1. According to FIA_UID.2, users must identify themselves before any action. FMT_MTD.1 determines who may access or manipulate TSF data. FMT_SMR.2 identifies the roles, links authorized users to each role, and ensures that the rules associated with the relationship between the users and the roles are met. FPT_STM.1 requires that reliable time stamps be provided by the TSF for TSF functions. FTA_TAB.1 provides access banners authorized by administrators for users before any session. FTA_TAH.1 provides the user with information about the previous successful and unsuccessful attempts to access the TOE under that claimed user identity.

O.CRYPTO_OPNS: The TOE will support cryptographic functions in a secure manner. User access to cryptographic IT assets will be restricted in accordance with a specified user access control policy. The TOE will provide one or more roles to manage cryptographic assets and attributes. There will be complete separation provided between plaintext and encrypted data and between data and keys. This requires separate channels and separate storage areas for data and keys. The only place any data can pass between the plaintext and encrypted data modules is in the cryptographic engine. There should be no way for plaintext keys to reach the data module and no way for data to enter the key-handling module. Encrypted keys can be handled as encrypted data, but with limited user access. The TOE will protect cryptographic data assets when they are being transmitted to and from the TOE, either through intervening untrusted components or directly to/from human users. The TOE will prohibit the transmission of a ciphertext message over internal circuitry where the corresponding plaintext might be available. To protect message metadata, the TOE will support cryptographic padding (e.g. random plaintext) and

encrypted addressing. The cryptographic components, functions, and interfaces shall be fully defined to ensure that the cryptographic keys have appropriate protection throughout their lifecycle, including generation, distribution, storage, use, and destruction. There will be self-tests, as well as alarms, alarm checks, and redundant logic, to provide the ability to verify that the cryptographic functions operate as designed. The TOE will produce, through robust encryption techniques, cipher text that cannot be decrypted without either massive computational power or knowledge of the encryption key.

Coverage Rationale: FCS_CKM.1 requires cryptographic keys to be generated according to an assigned specific generation algorithm. Cryptographic key distribution is controlled by FCS_CKM.2, which provides for a specified distribution method. The access to cryptographic keys is specified in FCS_CKM.3. The TSF shall securely erase classified and sensitive information, including cryptographic keys, in accordance with a specified cryptographic key destruction method (FCS_CKM.4) to ensure that compromise of keys is minimized. FCS_COP.1 sets standards requirements for the list of cryptographic operations, including cryptographic message padding and, possibly, user-set and random timing delays and encrypted addressing. FDP_ACC.1 enforces access control policies on subjects and objects subject to the access control policies. The TSF will enforce access control policies based on the security attributes covered in FDP_ACF.1. FDP_IFC.1 requires that an information flow control policy be set for each identified information flow in the TOE. Security attributes are required to secure information and subjects that collect the information (FDP_IFF.1). FDP_IFF.5 prohibits illicit information flows. Data storage shall be monitored by the TOE to identify integrity errors (FDP_SDI.1). FMT_MTD.1 restricts management of TSF data to authorized users assuming an authorized role defined in FMT_SMR.2 which also links users to their roles and enforces restrictions on the roles. Access to TOE resources will be controlled by the security management, which specifies roles for security purposes in addition to restrictions that specify the relationship between the roles (FMT_SMR.2). FPT_SEP.3 ensures the TSF has its own domain that is maintained separate and distinct to prevent tampering and interference. FPT_TST.1 provides for periodic and condition-based self-testing by authorized users to ensure proper operation of the TOE and integrity of TSF data.

O.Data_Exchange_Conf: The TOE will protect user data confidentiality when exchanging data with a remote system.

Coverage Rationale: FDP_ETC.1 and FDP_ITC.1 allow the export and import, respectively, of user data without security attributes as long as the access control and information flow control policies are followed. FDP_UCT.1 protects any exchange of data being transferred through a system in the TOE.

O.FAULT_TOLERANCE: The TOE will provide fault tolerant operations for critical components and continue to operate in the presence of specific failures in one or more system components. The TOE will automatically recover to a secure state without security compromise after system error or other interruption of system operation. The TOE will preserve the secure state of the system, as well as the level of assurance of the system, in the event of a secure component failure.

Coverage Rationale: FPT_FLS.1 will preserve a secure state when certain listed types of failures occur. FPT_RCV.1 requires a manual recovery function that will ensure the TOE returns to a secure state. Manual recovery allows the administrator to evaluate the cause of the failure and remove it. FPT_RCV.4 calls for an automatic recovery function that will ensure the TOE automatically returns to a secure state after a power loss without undue loss of data. FRU_FLT.2 requires the TOE to continue operating in a secure manner when certain failures occur.

O.IDENTIFY: The TOE will uniquely identify and authenticate each user of the system to support accountability through basic I&A functions. The TOE will associate each user-requested action with the identity of the user who initiated the session (i.e., log on).

Coverage Rationale: In the event that the number of consecutive logon failure exceeds the predetermined threshold, FIA_AFL.1 ensures session lockout, requiring administrator reset of the TOE. FIA_ATD.1 allows security attributes of the users to be managed individually. FIA_UAU.2 requires authentication for all users before any TSF-regulated action can be performed. FIA_UAU.7 ensures that only a limited amount of information be provided during authentication. FIA_UID.2 requires identification authentication for all users before any TSF-regulated action can be performed. All security attributes for each user are associated with subjects acting on behalf of the user (FIA_USB.1). FPT_STM.1 provides time stamps that are reliable for all TSF functions. FTA_TAB.1 provides access banners authorized by administrators be displayed before any session.

O.INTEGRITY: The TOE will provide the following technical features to protect its system security functions: detect changes to its security-related functions and user data, protect against tampering by users, and protect against attempts by users to bypass its security functions. The TOE will provide the ability for authorized users to verify that the system operates as designed, to conduct periodic integrity checks on both system and user data, and to conduct periodic system functional tests to test the integrity of the hardware and code running system functions. The TOE will always invoke mechanisms that enforce security policies. It will maintain at least one security domain for system (TOE) execution to protect the TOE from interference and tampering. Likewise, it will ensure that security-relevant software, hardware, and firmware are correctly functioning through features and procedures. The TOE will provide system features that detect physical tampering of a system component and will use those features to limit security breaches. The TOE will send integrity data, results of the integrity checks, to the audit trail. Additionally, it will prevent or resist physical tampering with specified system devices and components.

Coverage Rationale: FDP_SDI.1 requires that the SF monitor user stored data within the TSF to identify integrity errors. FDP_UIT.1 protects user data integrity transfer by detection of modifications. FMT_MSA.2 ensures that security attributes have secure values. FPT_AMT.1 provides for the ability to have the user initiate or condition-based testing of the abstract machine underlying the TSF. FPT_PHP.1 and FPT_PHP.3 require the TOE to detect and prevent, respectively, tampering with TSF functions. Protection

from physical attack or tampering of the TOE is covered by. FPT_RVM.1 ensures that the security functions of the TOE are always invoked and cannot be bypassed. FPT_SEP.3 requires that the TSF shall maintain separation and distinction between security domains and protects them from untrusted subjects. FPT_TST.1 requires the ability to test the TOE to ensure the correct operation and verify the integrity of TSF data and codes.

O.Integrity_Attr_Exch: The TOE will ensure that the system correctly exchanges security-attribute information with another trusted IT product.

Coverage Rationale: FPT_TDC.1 ensures the system correctly exchanges security–attribute information with another trusted system.

O.MANAGE: The TOE will provide adequate management features for its security functions. It will maintain security-relevant roles and the association of users with those roles. In addition to user identity, the TOE will maintain a set of security attributes associated with individual users. The TOE will provide features to specify object classes (domains), user groups, and operation classes. The management features will control what users can do in a given group by specifying which users may perform certain operations on particular objects.

Coverage Rationale: FDP_ACC.1 enforces access control policies on subjects and objects subject to the access control policies. The TSF will enforce access control policies based on the security attributes covered in FDP_ACF.1. FIA_ATD.1 allows security attributes of the users to be managed individually. FIA_USB.1 ensures security attributes for each user are associated with subjects acting on behalf of the user. FMT_MSA.3, through static attribute initialization, ensures default values for security attributes will be either permissive or restrictive in nature. FMT_SMR.2 maintains security roles, links users with the roles, and rules on the roles and users. FMT_SMR.3 requires that explicit requests for a particular roles be made to the TSF before the role may be assumed.

O.No_Residual_Info: The TOE will ensure there is no "object reuse," i.e., ensure that there is no residual information in some information containers or system resources upon their reallocation to different users.

Coverage Rationale: FDP_RIP.2 in the TSF ensures that there is no remaining information in systems upon their reallocation.

O.Session_Termination The TOE will lock and then terminate a session after a given interval of inactivity.

Coverage Rationale: FTA_SSL.1 and FTA_SSL.3 ensure that the TSF will lock and then terminate the session, respectively, after determined periods of inactivity.

6.2.1.2 – Environmental Objectives Rationale

Table 5 - Functional Component to Environmental Objective Mapping

Environmental Objective	Requirements
OE.ADMIN	FIA_SOS.1, FMT_MOF.1, FMT_MSA.1, FMT_MTD.2, FMT_MTD.3, FMT_REV.1, FMT_SAE.1, FMT_SMR.2, AGD_ADM.1, AGD_USR.1, ALC_DVS.2, ALC_FLR.3, ALC_LCD.2, ALC_TAT.3, AMA_AMP.1, AMA_CAT.1, AMA_EVD.1, AMA_SIA.2
OE.AUDIT_MAINTAIN	FAU_SAR.3, FAU_STG.2, FAU_STG.3, FAU_STG.4
OE.BANNER	FTA_TAB.1
OE.CRYPTO_DESIGN	NITR, ADV_FSP.3, ADV_HLD.4, ADV_INT.1, ADV_LLD.2, ADV_RCR.2, ADV_SPM.1, AGD_ADM.1, AGD_USR.1, ALC_TAT.3, ATE_COV.3, ATE_DPT.3, ATE_FUN.2, ATE_IND.2, AVA_VLA.4,
OE.DOCS	AGD_ADM.1, AGD_USR.1
OE.Malicious_Code	FDP_ITC.1, FPT_AMT.1, FPT_PHP.1, FPT_TST.1
OE.OPERATE	NITR, AGD_USR.1
OE.Screen_Lock	FTA_SSL.2
OE.Source_Code_Exam	NITR, ADV_IMP.3, ADV_LLD.2, ADV_RCR.2

OE.ADMIN: Administrators manage the TOE in a manner that maintains the system security. While the TOE is in operation, the administrator will control access to the system by maintenance personnel who troubleshoot the system and perform system updates. To securely manage the TOE, the administrator should know the origin of all data files and executables that the TOE, remote host, and secure enclave may generate, store, process, transmit, or receive. Administrators will terminate maintenance user system access privileges after expiration of an assigned timed interval. The administrator will also manage the initialization of, values for, and limits on allowable operations on security attributes, security critical data, and security mechanisms. The administrator, using the security tools and techniques employed during the development phase, will detect and resolve flaws during the operational phase and document the flaws. When TOE hardware, software, or firmware must be destroyed, the administrator will employ safe destruction techniques. Administrators will apply code fixes to fix the code when there are known security vulnerabilities in the code. This is particularly important with respect to the operating system. The administrator will implement a configuration management plan to assure storage integrity, identify system connections, and identify the system components (software, hardware, and firmware). Part of configuration management is ensuring that integrity data is not lost or misplaced. Any circumstances that can cause untrusted recovery will be documented with mitigating procedures established. Configuration management is critical to maintaining certification to operate the TOE. The administrator will

manage and update user authorization and privilege data, system security policy data, enforcement functions, and other security-relevant configuration data in accordance with organizational security policies. Administrators are responsible for the proper disposal of user data after access removal (due to job termination etc.). The administrator will manage resource security attributes and security-critical (TSF) data to ensure that the size of the data does not exceed the space allocated for storage of the data. The administrator will communicate system threats and vulnerabilities to system stakeholders.

Coverage Rationale: FIA_SOS requires that administrators develop metrics for the quality of secrets in the TOE. The TSF verifies secrets that meet particular metrics of the system. FMT_MOF.1 requires administrators to declare which roles may manage TSF functions. Administrators enforce TSP on the various roles in FMT_MSA.1. Administrators will determine the limits of TSF data and the actions to be taken when the data limit is approached or exceeded in FMT_MTD.2. FMT_MTD.3 ensures that the data has secure values. In FMT_REV.1, the administrator will identify the roles that can revoke security attributes and the revocation rules. FMT_SAE.1 provides for specification of an expiration date on security attributes and specification of the roles authorized to determine what action to take upon security attribute expiration. FMT_SMR.2 requires the administrator to identify the roles in the TOE. AGD_ADM.1 and AGD_USR.1 provide guidance from the developers for administrators and users, respectively, to use and administer the TOE. ALC_DVS.2 requires developers to identify security measures and show their sufficiency to help administrators maintain security through the lifetime of the TOE. ALC_FLR.3 requires the developer to provide at least one point-of-contact to work with the administrators and users to remove security flaws from the TOE. ALC_LCD.2 provides a model of the TOE to minimize flaws in the TOE system during development phase. ALC_TAT.3 requires the developer to define the tools and implementation options and standards used to develop the TOE. AMA_AMP.1 requires the developer to create plans to assure that the certification of the TOE is maintained after initial release. Administrators will have a major input to this plan. Along with the AMA_AMP.1, AMA_CAT.1, categorizes components of the system according to their importance and for re-evaluation of the TOE. Assurance of system security requirements is maintained by the developer and assessed by an evaluator covered by AMA_EVD.1. Administrators will work with the developers to evaluate the system security impact of configuration changes to the TOE in AMA_SIA.2.

OE.AUDIT_MAINTAIN: Administrators will apply technical, procedural, and administrative controls that are sufficient to maintain user accountability throughout the TOE. An audit-administration role will be created. The audit administrator will define the system response to possible loss of audit records when audit trail storage is full or nearly full; protect audit records against unauthorized access, modification, or deletion to ensure accountability of user actions; maintain audit data, guarantee space for that data, and regularly review audit data. The administrator will communicate anomalous audit data to system stakeholders.

Coverage Rationale: FAU_STG.2 guarantees availability of auditable data by TSF maintenance when a secure violation exists. FAU_STG.3 specifies certain actions to be taken when the threshold on an audit trail reaches or exceeds its limit. FAU_STG.4 prevents audit data loss in the event of a full audit trail.

OE.BANNER: The system will provide a banner to notify all users that they are entering a government or business computer system and their actions will be audited. Consequently, the banner informs the user of the possibility that the system will monitor user actions and that misuse of the system may result in criminal, civil, or administrative penalties.

Coverage Rationale: FTA_TAB.1 provides a display of a TOE access banner prior to the use of a secure system.

OE.CRYPTO_DESIGN: objective implements the engineering design requirements that DoD imposes on cryptosystems. The developer shall fully define cryptographic components, functions, and interfaces; minimize, or even eliminate, design and implementation errors in the cryptographic modules and functions, and prevent errors in one part of the TOE from influencing other parts, especially cryptographic parts. To this end, non-cryptographic input/output paths must be well defined and logically independent of circuitry and processes performing key generation, manual key entry, key erasure, and similar key-related operations. The developer shall specify cryptographic security functional requirements (SFRs) that are expected to be handled by other software, hardware, or firmware that is external to the TOE. The developer shall test cryptographic operation and key management functions.

Coverage Rationale: This is a non-information technology requirement (NITR) because all of the requirement components are assurance requirements that do not implement functions in the TOE. ADV_FSP.3 requires semiformal functional specifications describing external interfaces in a semiformal manner with informal language support. High level design, ADV_HLD.4, provides assurance that the TOE development process identifies and justifies the mechanisms used to separate TSP and non-TSP functions. ADV_INT.1 requires development of a modular TOE. ADV_LLD.2 provides assurance that the low-level design of the TOE is functionally defined in a semiformal manner. ADV_RCR.2 requires that all relevant TSF representations be semiformally specified and that adjacent abstract representations be consistent with each other. ADV_SPM.1 requires that the security function specification of policies in the TSP be modeled

semiformally. AGD_ADM.1 and AGD_USR.1 provide guidance from the developers for administrators and users, respectively, to use and administer the TOE. ALC_TAT.3 requires the developer to define the tools and implementation options and standards used to develop the TOE. ATE_COV.3 provides a rigorous method of analyzing TSF security functions to ensure security of the system. ATE_DPT.3 requires testing of the TOE implementation. ATE_FUN.2 provides documentation of proper testing by the developer to ensure all security functions perform as expected. ATE_IND.2 tests a sample of the security functions by independent evaluators. Test procedures are performed to ascertain vulnerabilities of the system and evaluate if they can be exploited in the TOE in AVA_VLA.4.

OE.Malicious_Code: Administrators will incorporate malicious code prevention procedures and mechanisms.

Coverage Rationale: TSF functions will be verified by self-test to ensure proper operations of data integrity (FPT_AMT.1).

OE.OPERATE: Authorized users and administrators will operate the TOE in a manner that maintains the system security by following adequate guidance documentation. Documentation provided to them will detail the proper use of the TOE to minimize the security risks within the environment.

Coverage Rationale: This is a NITR. AGD_ADM.1 and AGD_USR.1 provide guidance from the developers for administrators and users, respectively, to use and administer the TOE.

OE.Screen_Lock: The operating system or an application will provide a screen lock function to prevent an unauthorized user from using an unattended computer where a valid user has an active session.

Coverage Rationale: FTA_SSL.2 allows the user to lock or unlock their own interactive sessions to prevent unauthorized access.

OE.Source_Code_Exam: The developer, an independent tester, or an administrator (or a combination of all three parties) will examine source code for accidental or deliberate flaws in code made by the developer. The accidental flaws could be lack of engineering detail or bad design. Deliberate flaws would include building trapdoors for later entry.

Coverage Rationale: This is a NITR. ADV_IMP.3 requires that the evaluator determine the accuracy of TOE security functional requirements. ADV_LLD.2 provides assurance that the low-level design of the TOE is functionally defined in a semiformal manner. ADV_RCR.2 requires that all relevant TSF representations be semiformally specified and that adjacent abstract representations be consistent with each other.

6.2.3 - Assurance Security Requirements Rationale

The EAL for the HCT TOE is EAL 5, augmented.

The information that a remote user sends to and from the secure enclave may be For Official Use Only or highly classified and sensitive. The nature of the information makes it a highly sought after target for extremely capable adversaries. The physical environment for the remote user, Remote Unit, and the communications network that the information traverses are expected to be outside the controls of the remote user and HARA system customer. Therefore, there are many opportunities for the adversary to try to access the information either through the network or through direct presence near the remote user. Please note that TEMPEST requirements are not part of the Common Criteria, nor are they part of this profile, but TEMPEST might reasonably be expected to add value to the information security of the HARA system's remote unit.

For high-assurance TOEs, DoD policy[1] requires an EAL "greater than EAL 4" [4]. EAL 5 meets these criteria and provides a full measure of assurance that accompanies a CC-defined EAL.

For the environment described for this TOE, threat agents are sophisticated and the information protected by the TOE is very sensitive. With such an environment, it is reasonable to expect significant expense in the area of security engineering of a product. A relatively higher degree of security technology engineering is expected to be applied for a TOE in this environment. Furthermore, the system engineering practices required to coordinate the development and integration of multiple components that have high security engineering functionality, such as this HCT TOE, into an adaptable system is unlikely at the level of engineering practice required at EAL 4. EAL 5 provides value by specifying semi-formal presentation of the functional specification and high-level design and semi-formal demonstration of correspondence between them. EAL 5 also calls for a formal model of the TOE security policy (TSP) and a modular design.

The TOE described in this PP is a component that will be integrated into a larger system that will be produced by more than one development organization. The case of independent integration is highlighted because this as a worst-case scenario when compared to a single organization building (and integrating internally) all the components to form the system. Each individual development organization may have its own degree system security engineering practices. The integrator of a collection of TOEs into a cohesive system will depend on each TOE vendor making the interfaces and functionality clear through a semi-formal method.

Many of the assurances selected match the requirements for EAL 6. EAL 6 was not selected because ADV_INT.1: Modular Design was considered the appropriate specification for the HCT PP component. By the rules of the Common Criteria, the TOE can be said to meet the highest EAL for which it meets all the assurances of the EAL. However, because of the nature of the information being processed, ALC_FLR.3:

[1] Global Information Grid (GIG) Policy 6-8510, Information Assurance Guidance, 16 June 2000.

Systematic Flaw Remediation, ATE_DPT.3: Testing Implementation Representation, and AVA_CCA.3: Exhaustive Covert Channel Analysis were selected. The Maintenance Assurance family was selected so that the requirements of the DoD Information Technology Security Certification and Accreditation Process (DITSCAP) could be referenced and met as early in the development as possible.

The EAL of the HARA system overall is EAL 5, but not all of the components of the system will actually process security data or classified information. The PP authors believe that an augmented or higher EAL at the component level is justified if the component, such as the HCT, handles security data or classified information. If problems arise in the future of the HARA system, the documentation of the development, installation, and test of the augmented of higher EAL components, such as the HCT, should direct the system security engineers' attention to the un-evaluated portions of the system, such as the Remote Unit operating system.

MINIMUM SOF ARGUMENTS

The strength-of-function (SoF) claim for this PP is SoF-High. This claim is based upon the fact that the TOE will process classified information related to national security. The application (remote access) and the distributed nature of TOE components imply that threat agents could have complete access to the TOE for an extended period of elapsed time.

The TOE objectives of this PP must enforce TOE policies and counter TOE-relevant threats with a degree of effectiveness that is commensurate with the threat posed. The risk environment is comprised of a very sophisticated threat agent in conjunction with very sensitive data. Thus, a rating of SoF-High is consistent with the TOE objectives included in this PP

6.3 - Dependency Rationale

Table 6 - Functional and Assurance Requirements Dependencies

Requirement	Dependencies
Functional Requirements	
FAU_GEN.1	FPT_STM.1
FAU_GEN.2	FAU_GEN.1, FIA_UID.1
FAU_SAR.3	-
FAU_SEL.1	FAU_GEN.1, FMT_MTD.1
FAU_STG.2	FAU_GEN.1
FAU_STG.3	FAU_STG.1
FAU_STG.4	FAU_STG.1
FCS_CKM.1	FCS_CKM.2, FCS_CKM.4, FCS_COP.1, FMT_MSA.2
FCS_CKM.2	FCS_CKM.1, FCS_CKM.4, FDP_ITC.1, FMT_MSA.2
FCS_CKM.3	FCS_CKM.1, FCS_CKM.4, FDP_ITC.1, FMT_MSA.2
FCS_CKM.4	FCS_CKM.1, FDP_ITC.1,FMT_MSA.2
FCS_COP.1	FCS_CKM.1, FCS_CKM.4, FDP_ITC.1, FMT_MSA.2
FDP_ACC.1	FDP_ACF.1
FDP_ACF.1	FDP_ACC.1, FMT_MSA.3
FDP_ETC.1	FDP_ACC.1, FDP_IFC.1
FDP_ETC.2	FDP_ACC.1, FDP_IFC.1
FDP_IFC.1	FDP_IFF.1
FDP_IFF.1	FDP_IFC.1, FMT_MSA.3
FDP_IFF.5	AVA_CCA.3, FDP_IFC.1
FDP_ITC.2	FDP_ACC.1, FDP_IFC.1, FPT_TDC.1
FDP_RIP.2	
FDP_SDI.1	
FDP_UCT.1	FDP_ACC.1, FDP_IFC.1
FDP_UIT.1	FDP_ACC.1, FDP_IFC.1,
FIA_AFL.1	FIA_UAU.1
FIA_ATD.1	
FIA_SOS.1	
FIA_UAU.2	FIA_UID.1
FIA_UAU.7	FIA_UAU.1
FIA_UID.2	
FIA_USB.1	FIA_ATD.1
FMT_MOF.1	FMT_SMR.1
FMT_MSA.1	FDP_ACC.1, FDP_IFC.1, FMT_SMR.1

Requirement	Dependencies
Functional Requirements	
FMT_MSA.2	ADV_SPM.1, FDP_ACC.1, FDP_IFC.1, FMT_MSA.1, FMT_SMR.1
FMT_MSA.3	FMT_MSA.1, FMT_SMR.1
FMT_MTD.1	FMT_SMR.1
FMT_MTD.2	FMT_MTD.1, FMT_SMR.1
FMT_MTD.3	ADV_SPM.1, FMT_MTD.1
FMT_REV.1	FMT_SMR.1
FMT_SAE.1	FMT_SMR.1, FPT_STM.1
FMT_SMR.2	
FMT_SMR.3	FMT_SMR.1
FPT_AMT.1	
FPT_FLS.1	ADV_SPM.1
FPT_PHP.1	FMT_MOF.1
FPT_PHP.3	
FPT_RCV.1	ADV_SPM.1, AGD_ADM.1, FPT_TST.1
FPT_RCV.4	ADV_SPM.1
FPT_RVM.1	
FPT_SEP.2	
FPT_SEP.3	
FPT_STM.1	
FPT_TDC.1	
FPT_TST.1	FPT_AMT.1
FRU_FLT.2	FPT_FLS.1
FTA_MCS.1	FIA_UID.1
FTA_SSL.1	
FTA_SSL.2	FIA_UAU.1
FTA_SSL.3	
FTA_TAB.1	
FTA_TAH.1	
FTA_TSE.1	

Requirement	Dependencies
Assurance Requirements	
ACM_AUT.2	ACM_CAP.3
ACM_CAP.5	ACM_SCP.1, ALC_DVS.2
ACM_SCP.3	ACM_CAP.3
ADO_DEL.3	ACM_CAP.3
ADO_IGS.2	AGD_ADM.1
ADV_FSP.3	ADV_RCR.1
ADV_HLD.4	ADV_FSP.3, ADV_RCR.2
ADV_IMP.3	ADV_INT.1, ADV_LLD.1, ADV_RCR.1, ALC_TAT.1
ADV_INT.1	ADV_IMP.1, ADV_LLD.1
ADV_LLD.2	ADV_HLD.3, ADV_RCR.2
ADV_RCR.2	
ADV_SPM.1	ADV_FSP.1
AGD_ADM.1	ADV_FSP.1
AGD_USR.1	ADV_FSP.1
ALC_DVS.2	
ALC_FLR.3	
ALC_LCD.2	
ALC_TAT.3	ADV_IMP.1
AMA_AMP.1	ACM_CAP.2, ALC_FLR.1, AMA_CAT.1
AMA_CAT.1	ACM_CAP.2
AMA_EVD.1	AMA_AMP.1, AMA_SIA.1
AMA_SIA.2	AMA_CAT.1
ATE_COV.3	ADV_FSP.1, ATE_FUN.1
ATE_DPT.3	ADV_HLD.2, ADV_IMP.2, ADV_LLD.1, ATE_FUN.1
ATE_FUN.2	
ATE_IND.2	ADV_FSP.1, AGD_ADM.1, AGD_USR.1, ATE_FUN.1
AVA_CCA.3	ADV_FSP.2, ADV_IMP.2, AGD_ADM.1, AGD_USR.1
AVA_MSU.3	ADO_IGS.1, ADV_FSP.1, AGD_ADM.1, AGD_USR.1
AVA_SOF.1	ADV_FSP.1, ADV_HLD.1
AVA_VLA.4	ADV_FSP.1, ADV_HLD.2, ADV_IMP.1, ADV_LLD.1, AGD_ADM.1, AGD_USR.1

6.4 - Security Functional Requirements Grounding in Objectives

Table 7 - Requirements to Objectives Mapping

Requirements	Objectives
ACM_AUT.2	None
ACM_CAP.5	None
ACM_SCP.3	None
ADO_DEL.3	None
ADO_IGS.2	None
ADV_FSP.3	OE.CRYPTO_DESIGN
ADV_HLD.4	OE.CRYPTO_DESIGN
ADV_IMP.3	OE.Source_Code_Exam
ADV_INT.1	OE.CRYPTO_DESIGN
ADV_LLD.2	OE.CRYPTO_DESIGN, OE.Source_Code_Exam
ADV_RCR.2	OE.CRYPTO_DESIGN, OE.Source_Code_Exam
ADV_SPM.1	OE.CRYPTO_DESIGN
AGD_ADM.1	OE.ADMIN, OE.CRYPTO_DESIGN, OE.DOCS
AGD_USR.1	OE.ADMIN, OE.CRYPTO_DESIGN, OE.DOCS, OE.OPERATE
ALC_DVS.2	OE.ADMIN
ALC_FLR.3	OE.ADMIN
ALC_LCD.2	OE.ADMIN
ALC_TAT.3	OE.ADMIN, OE.CRYPTO_DESIGN
AMA_AMP.1	OE.ADMIN
AMA_CAT.1	OE.ADMIN
AMA_EVD.1	OE.ADMIN
AMA_SIA.2	OE.ADMIN
ATE_COV.3	OE.CRYPTO_DESIGN
ATE_DPT.3	OE.CRYPTO_DESIGN
ATE_FUN.2	OE.CRYPTO_DESIGN
ATE_IND.2	OE.CRYPTO_DESIGN
AVA_CCA.3	None
AVA_MSU.3	None
AVA_SOF.1	None
AVA_VLA.4	OE.CRYPTO_DESIGN
FAU_GEN.1	O.AUDIT
FAU_GEN.2	O.AUDIT

Requirements	Objectives
FAU_SAR.3	OE.AUDIT_MAINTAIN
FAU_SEL.1	O.AUDIT
FAU_STG.2	OE.AUDIT_MAINTAIN,
FAU_STG.3	OE.AUDIT_MAINTAIN
FAU_STG.4	OE.AUDIT_MAINTAIN
FCS_CKM.1	O.CRYPTO_OPNS
FCS_CKM.2	O.CRYPTO_OPNS
FCS_CKM.3	O.CRYPTO_OPNS
FCS_CKM.4	O.CRYPTO_OPNS
FCS_COP.1	O.CRYPTO_OPNS
FDP_ACC.1	O.CRYPTO_OPNS, O.MANAGE
FDP_ACF.1	O.CRYPTO_OPNS, O.MANAGE
FDP_ETC.1	O.Data_Exch_Conf
FDP_IFC.1	O.CRYPTO_OPNS
FDP_IFF.1	O.CRYPTO_OPNS
FDP_IFF.5	O.CRYPTO_OPNS
FDP_ITC.1	O.Data_Exch_Conf
FDP_RIP.2	O.No_Residual_Info
FDP_SDI.1	O.CRYPTO_OPNS, O.INTEGRITY
FDP_UCT.1	O.Data_Exchange_Conf
FDP_UIT.1	O.INTEGRITY
FIA_AFL.1	O.IDENTIFY
FIA_ATD.1	O.IDENTIFY, O.MANAGE
FIA_SOS.1	OE.ADMIN
FIA_UAU.2	O.ACCESS, O.IDENTIFY
FIA_UAU.7	O.IDENTIFY
FIA_UID.2	O.IDENTIFY, O.AUDIT
FIA_USB.1	O.IDENTIFY, O.MANAGE
FMT_MOF.1	OE.ADMIN
FMT_MSA.1	OE.ADMIN
FMT_MSA.2	O.INTEGRITY
FMT_MSA.3	O.MANAGE
FMT_MTD.1	O.AUDIT, O.CRYPTO_OPNS
FMT_MTD.2	OE.ADMIN
FMT_MTD.3	OE.ADMIN
FMT_REV.1	OE.ADMIN
FMT_SAE.1	OE.ADMIN

Requirements	Objectives
FMT_SMR.2	O.AUDIT, O.CRYPTO_OPNS, O.MANAGE, OE.ADMIN
FMT_SMR.3	O.MANAGE
FPT_AMT.1	OE.Malicious_Code, O.INTEGRITY
FPT_FLS.1	O.FAULT_TOLERANT
FPT_PHP.1	O.INTEGRITY
FPT_PHP.3	O.INTEGRITY
FPT_RCV.1	O.FAULT_TOLERANT
FPT_RCV.4	O.FAULT_TOLERANT
FPT_RVM.1	O.INTEGRITY
FPT_SEP.2	O.CRYPTO_OPNS, O.INTEGRITY
FPT_SEP.3	O.INTEGRITY
FPT_STM.1	O.AUDIT, O.IDENTIFY
FPT_TDC.1	O.Integrity_Attr_Exch
FPT_TST.1	O.CRYPTO_OPNS, O.INTEGRITY
FRU_FLT.2	O.FAULT_TOLERANT
FTA_MCS.1	O.ACCESS
FTA_SSL.1	O.Session_Termination
FTA_SSL.2	OE.Screen_Lock
FTA_SSL.3	O.Session_Termination
FTA_TAB.1	O.AUDIT, O.IDENTIFY, OE.BANNER
FTA_TAH.1	O.AUDIT
FTA_TSE.1	O.ACCESS

Appendix A - Acronyms

CC - Common Criteria
EAL - Evaluation Assurance Level
IT - Information Technology
PP - Protection Profile
SF - Security Function
SFP - Security Function Policy
SOF - Strength of Function
ST - Security Target
TOE - Target of Evaluation
TSC - TSF Scope of Control
TSF - TOE Security Functions
TSFI - TSF Interface
TSP - TOE Security Policy

References

[1] Draft U.S. DoD Remote Access Protection Profile for High Assurance Environments, version 0.98, 24 May 2000

[2] Common Criteria Implementation Board. Common Criteria for Information Technology Security Evaluation, Version 2.1. CCIMB-99-021, 032, 033. August 1999.

[3] High-Assurance Remote Access (HARA) Architecture, Version 1.1, 15 May 2000.

[4] Global Information Grid (GIG) Policy 6-8510, Information Assurance Guidance, 16 June 2000.